04/03/2025

Page No.
Date

✡ SQL — Structured query language.

Data - Data was a raw - fact which describes Attribute of an Entity

✡ Attribute — Properties

✡ Entity - Object

✡ Raw - fact - Original - truth

Entity :- Entity is nothing but an object it may be either living things or non - living things.

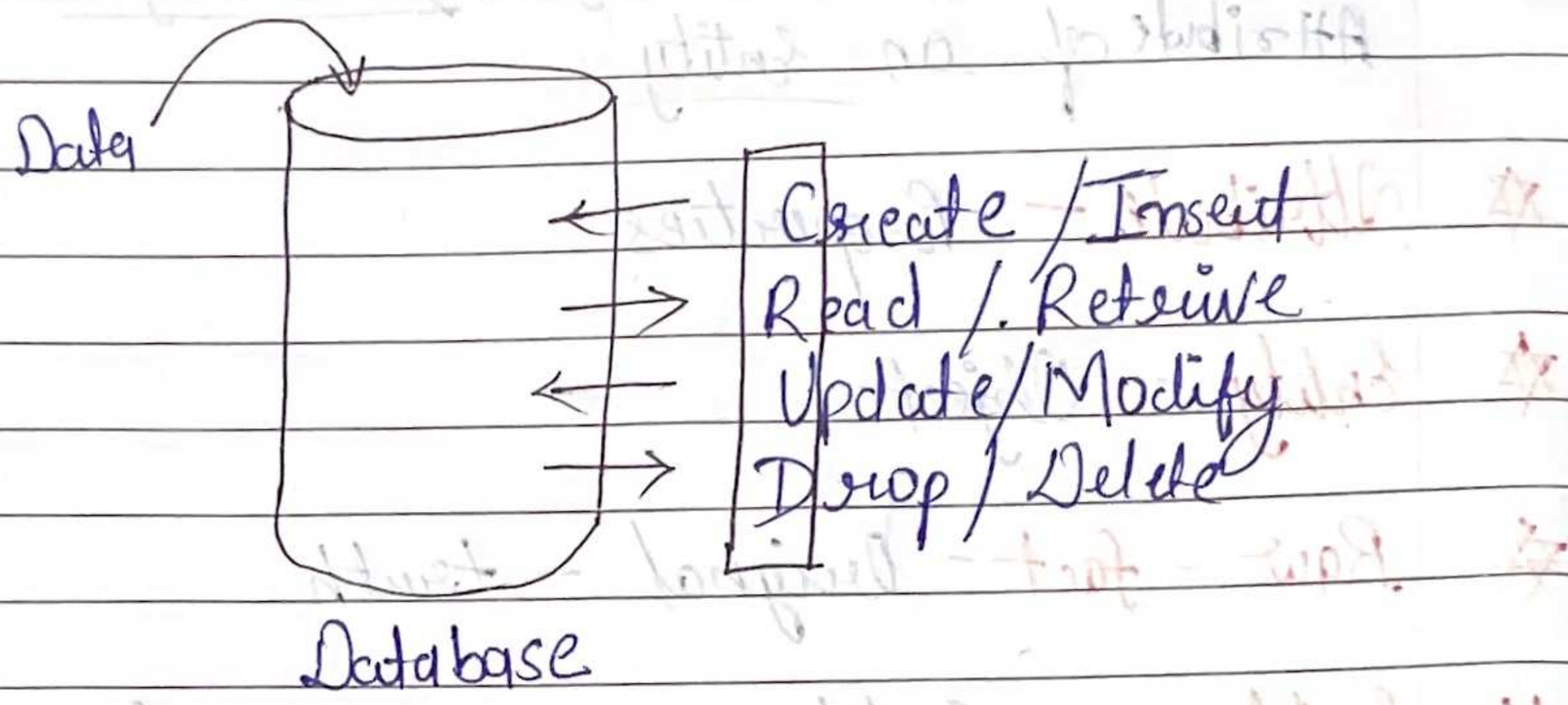
Any things which has physical existance in a real world is known a Entity.

✡ Database :- Database is a place, mediums, or a container which is used into store a data in a systematic and Organized manner.

In Database we can perform several operations.

1. Create / Insert
2. Read / Retrive
3. Update / Modify
4. Delete / Drop

This operations are universally reform as CRUD operations.



✡ DBMS – Database management System

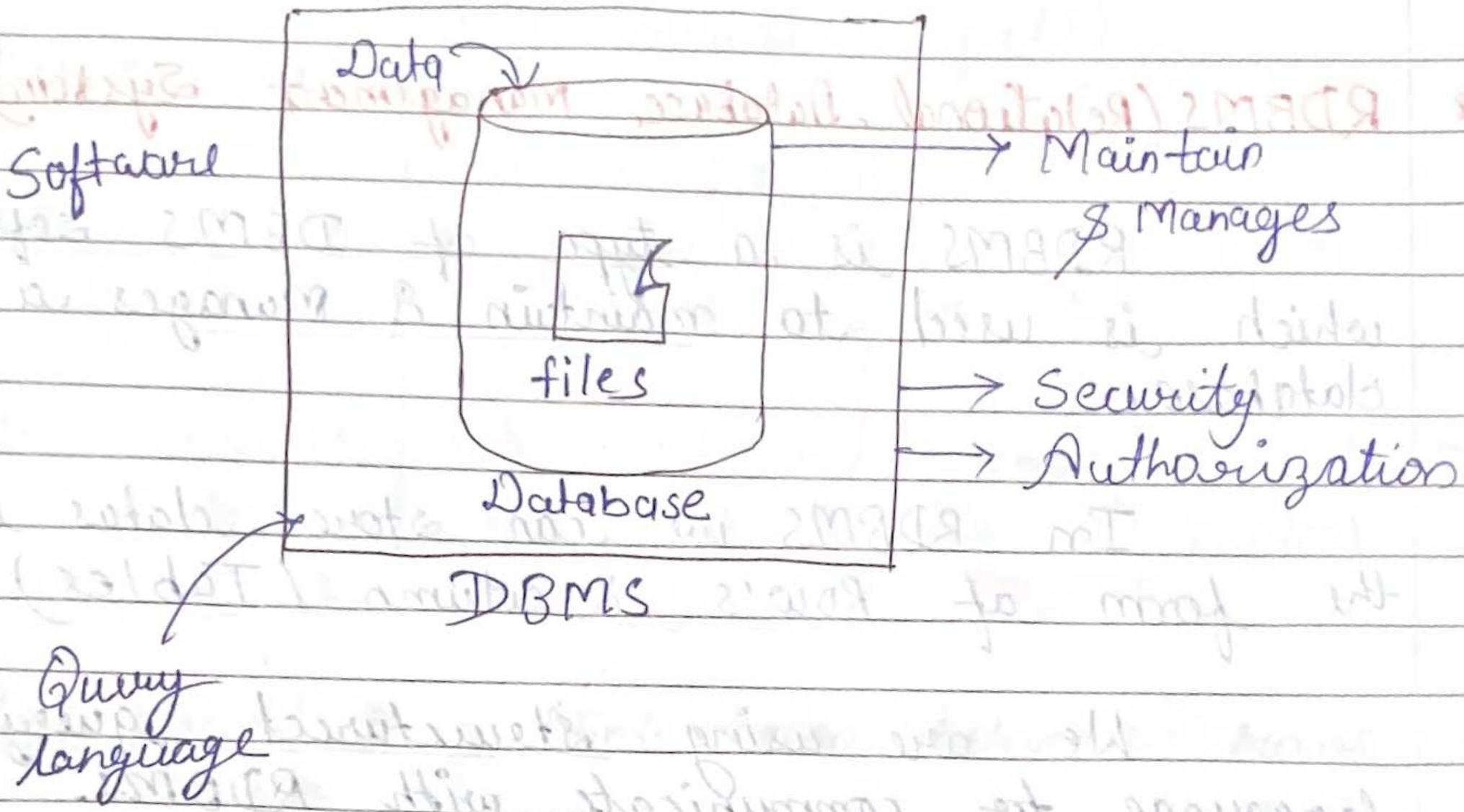
→ Database DBMS is a Software which is used to maintain & Manages the database.

It consists of two important features

1. Security
2. Authorization

→ In DBMS we can Store Data in the form of files.

→ We are using query languages to communicate with DBMS.



☆ Types of DBMS :-

1. Hierarical DBMS (HDBMS)
2. Network Database management system (NDBMS)
3. Object-oriented Database management System (OODBMS)
4. Relational Database management System (RDBMS)

Relational Model :- Relational model was designed by the data scientist E.F. Codd.

Any DBMS which supports relational model then it becomes RDBMS

[DBMS] + [Relational Model] → [RDBMS]

Any DBMS which follows the rules of E.F. Codd then it becomes RDBMS

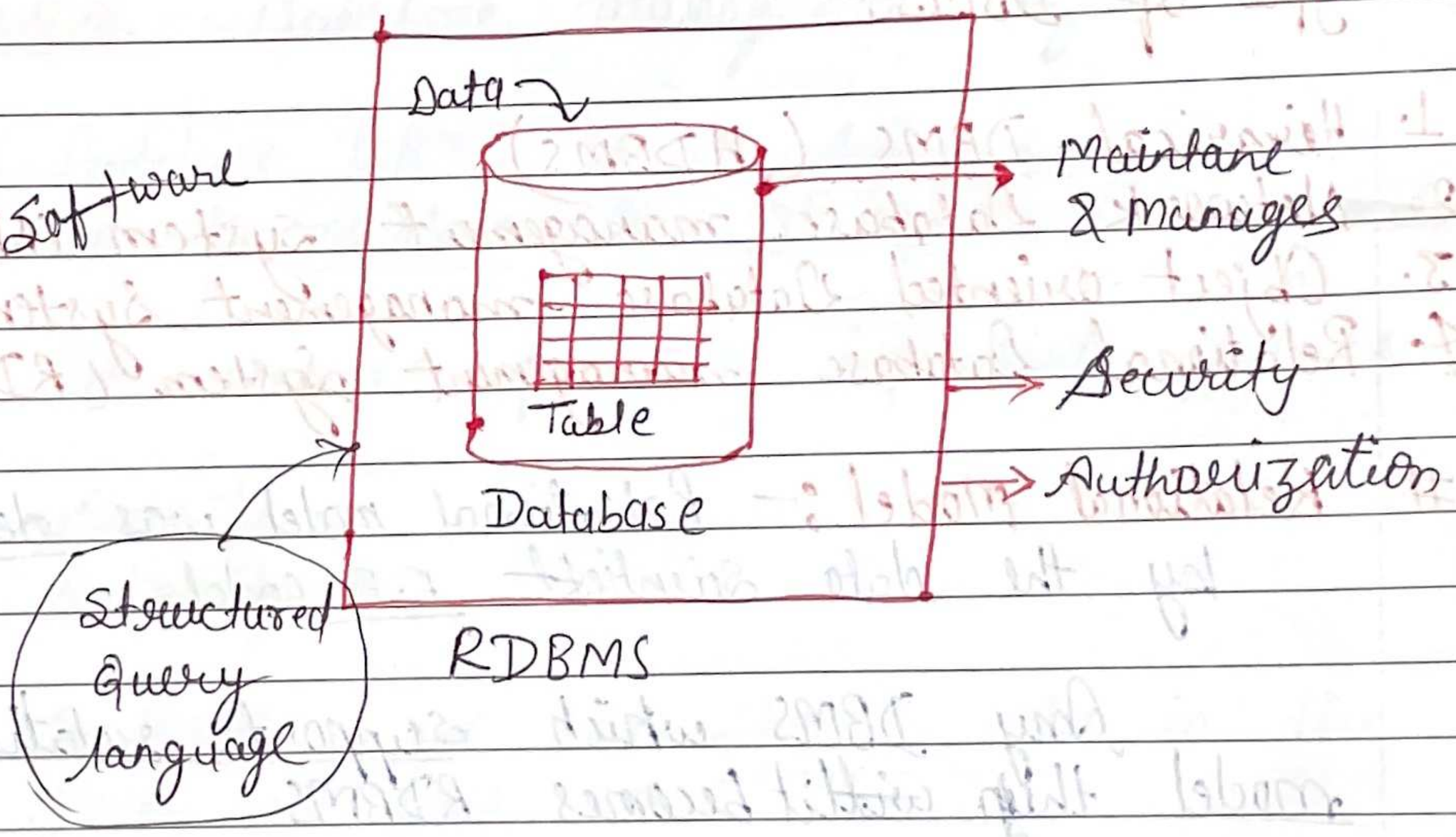
[DBMS] + [Rules of E.F. Codd] → [RDBMS]

✡ RDBMS (Relational Database Management System):-

RDBMS is a type of DBMS software which is used to maintain & manages a database.

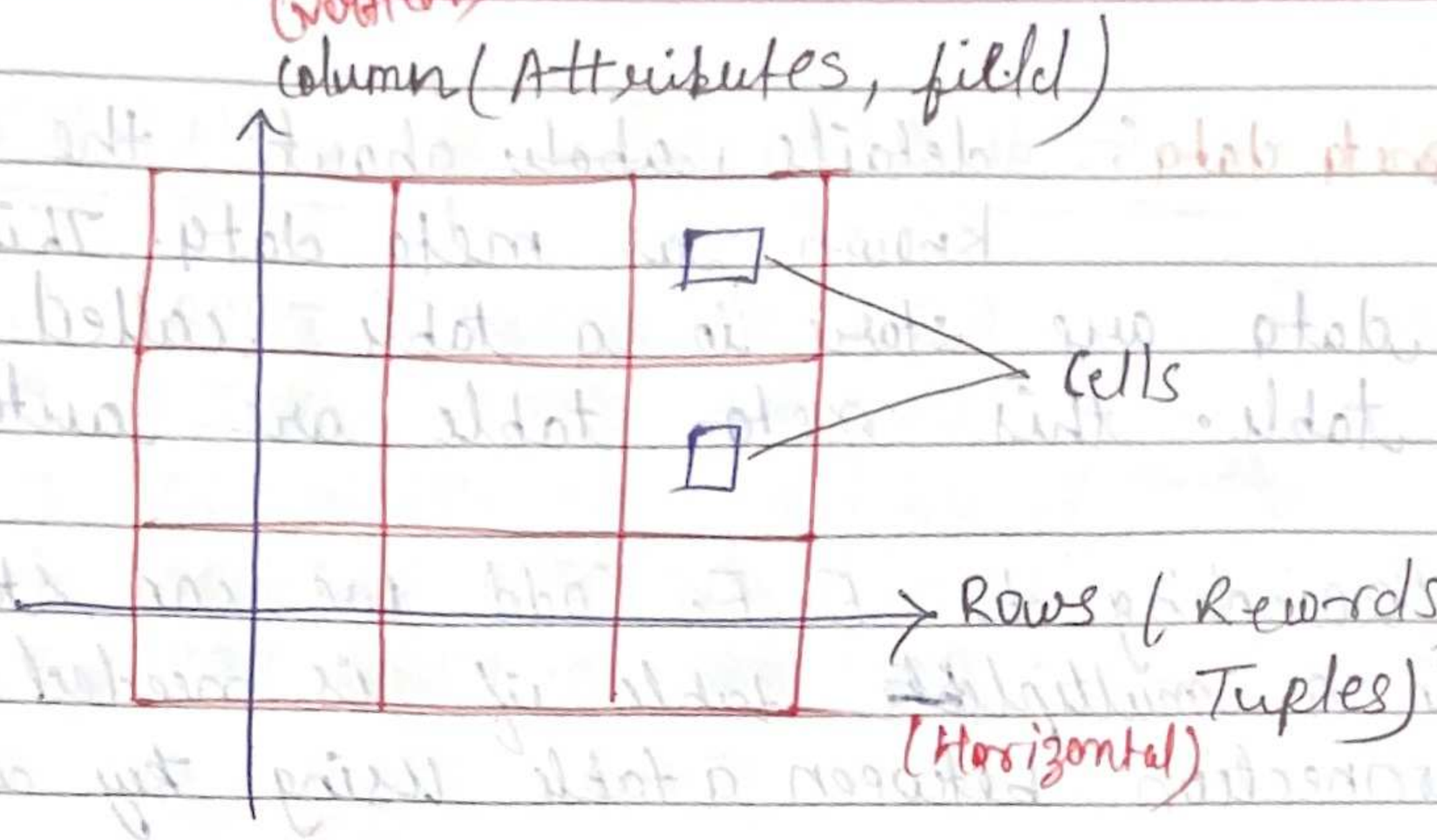
In RDBMS we can store datas in the form of Row's & column (Tables).

We are using structured query language to communicate with RDBMS.



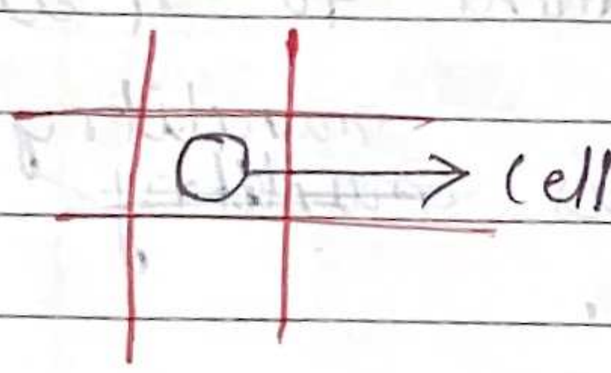
★ Table :- Table is a logical organization of Rows, column & cells.

Table:



★ Cell :- Cell is a Intersection between rows and column.

cell is a smallest unit or a block in table which is used to stores a data.



★ Rules of E.F. Codd :—

Rule 1. Data entering into a cell must be a single value data. ★

★ If it is multi-value we are facing data loss.

Rule 2. In RDBMS we can store every thing in the form of tables including meta data.

Page No.
Date

meta data :- details ~~about~~ about the data is known as meta data. This meta data are store in a table called meta table. this meta table are auto-generative.

Rule 3 According to E. F. Codd we can store datas in a multiple table if is needed Establish connection between a table using key attributes

Rule 4 In RDBMS we can validate the datas by two steps.

1. By Assning data types to a column

2. By Assning consteint to a column

Note :- Data type is ~~mandatory~~ mandating constaints are optional.

Page No.
Date

Employee

Photo	EID	Name	Phoneno	Salary	Dep No	Alternateno
	1	ABHI	9876543252	4000	30	
	2	SURYA	6379567255	7000	10	
	3	ARJUN	7892714581	8500	20	9875627356
	4	ARUN	6260257826	9000	10	
	5	RAM	6789752487	7000	30	

Relationship (key Attribute)

Data

Details of data

Meta data

Size : 1MP
Time : 12PM
Type : JPGG
Resolution : 1080

Size	Time	Type	Resolution
1MB	12PM	JPEG	1080

DataType

Constaint

DepNO.	Dname	L
10	D1	L1
20	D2	L2
30	D3	L3

Auto generated

★ Data Types :- It discribes which kind of type of data should be entering into perticular column or a memory allocation.

Types of datatype

1. char
2. Varchar / Varchar2
3. Date
4. Number
5. Large object
   - Charecter Large object
   - Binary large object

1. char :- [character]

→ 'A-z', 'a-z', '0-9', 'special characters (@, , , #, !, =, >>
'Alpha numeric'

→ Character are must inclosed within single qutation

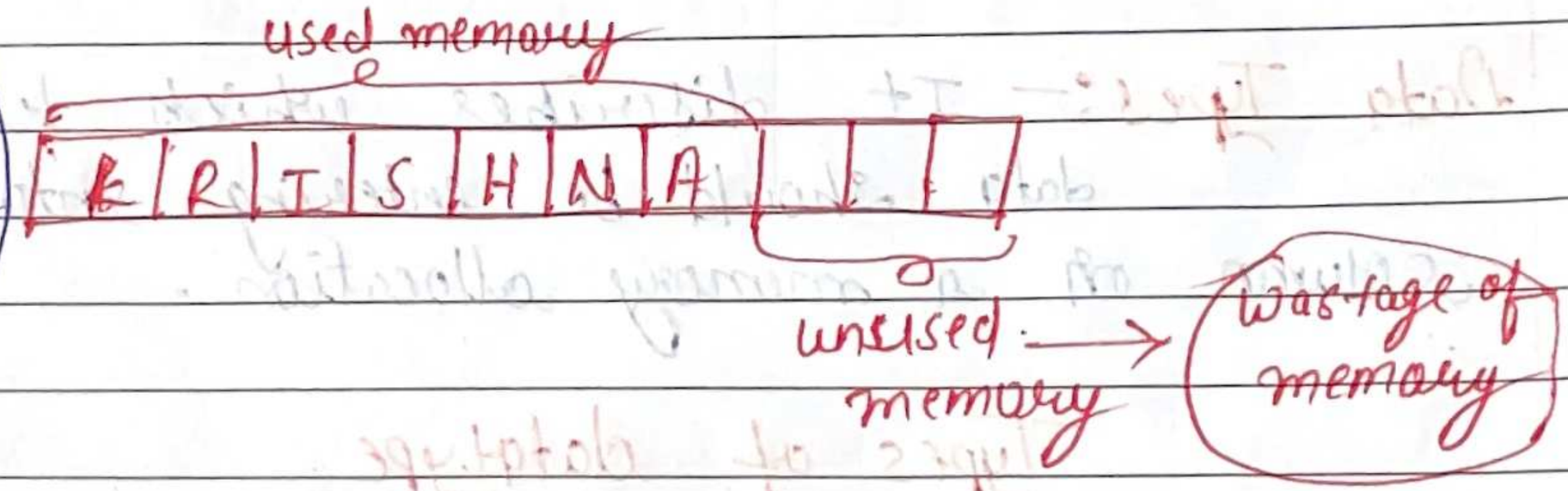
→ Syntax → char (size)

→ The Maximum size of char is upto 2000 charqcters

→ It follows fix length of memory allocation

ex: 'KRISHNA'

char (10)



2. Varchar :-

2. Varchar :- [Charater]

→ 'A-Z', 'a-z', '0-9', 'special character', 'Alpa numeric'
(@, &, #, !, \)

→ Single qeuotation

→ Syntax

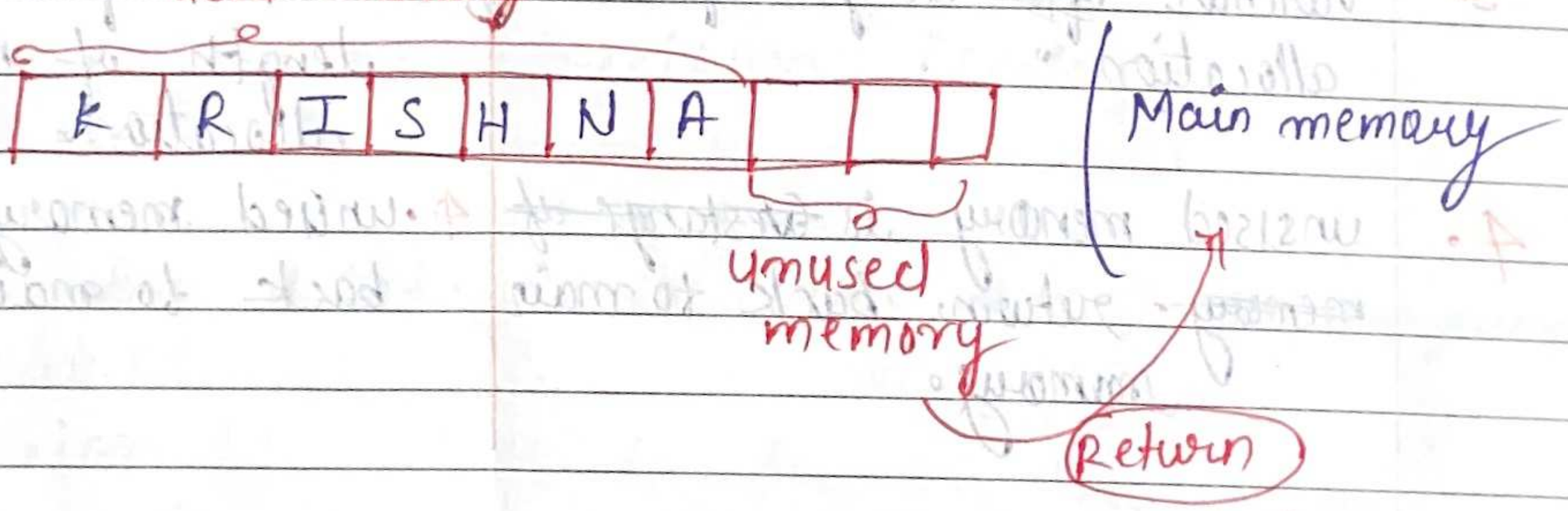
Varchar(size)

→ The maximum size of varchare was 2000 character

→ It follows variable length of memory allocation

eg: KRISHNA

Varchar (10)



3. Varchar 2 :- Updated version of Varchar.

→ Syntax

Varchar(size)
Varchar2(size)

→ The maximum size of varchar 2 was 4000 character.
(it follow same principle of varchar)

Page No.
Date

#Note :-

SQL is not a case sensitive language but character dates are case sensitive.

* Difference between Varchar & Varchar 2

	Varchar	Varchar 2
1.	Syntax :- Varchar (size)	1. Syntax :- Varchar (size) Varchar 2 (size)
2.	The maximum size of varchar is 2000 character.	2. The maximum size of Varchare 2 is 4000 Characters.
3.	Varchar follows Veriable allocation.	3. Varchar 2 follows Veriable length of memory allocation.
4.	unsised memory is ~~wastage of memory~~ return back to main memory.	4. unused memory is return back to main memory

Char

1. Syntax:
   char (size)
2. The maximum size of char is upto 2000 character
3. Char follows fix length of memory allocation.
4. unused memory is wastage of memory

4 Date :- It is used to store only the Date.

→ Syntax :-

Date

We can store dates in a oracle specified formate

'DD-MON-YYYY'

'01-JAN-2025'

(or)

'DD-MON-YY'

'01-JAN-25'

5. Number :- It is used to store only a numeric values. (0-9)

Syntax :-

Number (Precision, [scale])

→ In number datatype we can pass two arrgumen (inputs)

① precision

② scale

→ precision is maindtroy, scale is optional.

→ Precision - It is used to allocate number of digits requires to store an Integer values.

→ Range of precision is 1 to 38

→ Scale:- It is used to store the decimal values within the given precision.

→ The range of scale is -84 to 127.

→ The default value of scale is zero.

Eg: Number(5) : ± 9.999
↓P

Number(8) : ± 9.9999999
↓P

Number(5,3) : ± 99.999
↓P ↓S

Number(7,5) : ± 99.99999
↓P ↓S

Number(3,3) : ± .999
↓P ↓S

Number(3,5) : ± .00999
↓P ↓S

Number(4,8) : ± .0000 9999
↓P ↓S

6. Large Object:-

(i) character large objects:- It is used to store huge amount of character upto 4gb of size.

→ Syntax:- CLOB

(ii) Binary large object : It is used to store the binary values of photo, video, audio, document etc. upto 4 GB of size.

→ Syntax :- BLOB

Constaints :- Constaints are the rules or the condition which are assing to column for validation.

★ Types of constaints :-

1 UNIQUE
2 NOT NULL
3 CHECK
4 Primary key
5 ~~Foreign key~~ Foreign key

1. UNIQUE - Unique is constaint it can not accept duplicate or repeated values in a datas in column.

2. Not Null :- Not null is constaints it can't accept null values.

Null - Null is nothing but an empty cell.
→ Null can't occupies any memory space.
→ Zero or Space can't be a Null
→ If we can perform cany of the Arithmetic Task with null it returns null

3. CHECK :- CHECK is a constaint which used extra Validation with the help of conditions. If the condition get satisfied data entered into column or else it reject the data.

Check → [Extra Validation] → conditions

↳ check (condition)

[column-name operator value/Data]

Eg:

→ Check (salary >= 0)

7000 >= 0 → T

-9000 >= 0 → F

→ Check (Length (phone-no) = 10)

Length (9752166655) = 10

10 = 10

Length() ⇒ count the no. of character.

Length (phone-no)

4. Primary key :- It is used to identifing a unique records from the table.

Caracterstics of Primary Key.

1. We have only one primary key in table
2. Primary key can't accept duplicate or repeted values in a table

3. Primary key can't accept null values.
4. Primary key is a combination of unique and not null constraints.

→ Primary key is not ~~mandatory~~ mandatory but it highly ~~recommended~~ recommended to used in a table.

5. Foreign Key :- It is used to Establish connections between the table.

Characteristics of foreign key.

1. We have 'n' number of foreign keys in a table
2. foreign can accept duplicate or repeted values.
3. foreign key can accept null values.
4. foreign key is not a combination of unique and not null constraints.
5. for a attribute to become a foreign key it must must be primary key in it own table.
6. foreign key present in child table but actual belongs to parent table.
7. foreign key is also known referential integrity constraints.

Employee

column name →	EId	Name	PhoneNo	Salary	DepNO	CNO
Datatype →	Number(3)	Varchar(25)	Number(10)	Number (6,2)	Number (2)	Number (3)
Constraint →	Primary key	Not Null	Unique Not null Check: Length(PN)=10	Not null Check (salary >=0)	Foreign key	Foreign key
	101	ALLEN	9752116622	3200.00	20	12
	103	JONES	8752166655	9000.00		11
	108	SMITH	79521873	5110.00	10	
	105	JONES	9876548741	9720.00	20	12
	110	JAMES	9488772811	7910.00	10	11

Child - Table

F.K.

F.K.

Department

P.K. DepNo	Dname	Location
Number(2)	Varchar(20)	Varchar(35)
Primary key	Not null	Not Null
10	D1	L1
20	D2	L2

Parent table

Customer

P.K. Cno	Cname	Address
Number(3)	Varchar(20)	Varchar(35)
Primary key	Not null	Not Null
11	C1	A1
12	C2	A2

Parent table

✲ Overview of SQL

1. Data Defination language (DDL)
2. Data Manupulation language (DML)
3. Transaction control language (TCL)
4. Data Control language (DCL)
5. Data quary language (DQL)

1 Data quary language (DQL)

→ It is used to retrived or fatch dates from the ~~the table by selecting~~ database

✲ DQL Statement

1. Select → table → display
2. Projection → → column
3. Selection → → rows/column
4. Joins → multiple tables

① select :- It is used to retrieve the dates from the table and display.-

② Projection :- It is used to retrieve the dates from the table by selecting only the columns. If the ~~to~~ columns are get selected by default the values present in column also get selected.

③ Selection :- It is used to retrieve the datas from the table by selecting both row as well as columns.

4. Joins :- It is used to retrieve data from the multiple tables. (more than one one table) ~~Symentaniesly~~ Simultaneously.

✡ Projection :- It is used to retrieve data from the table by selecting only a column. If the columns get selected by default value present in column get selected.

Syntax :-

```
SELECT * /[DISTINCT] column_name/
    Expression [ALIAS] FROM Table_Name;
```

Q. Write a query to display all the Students names present in Student table

```
(2) select sname
(1) from student;
```

argument

output of from clause

Result table or output table

output of student clause

Database

Student

SId	Sname	Branch	Per.
1	A	B.Com.	80
2	B	EEE	90
3	C	MECH	75
4	D	AERO	80
5	E	CSE	70
6	B	IT	65

Sname
A
B
C
D
E
B

Under Execution

Page No.
Date

Note

① In projection - from clause ^ will execut ~~execut~~ first
② In from clause we can pass Table name as a argument.
3. The job of a from clause will goes to a database and makes the table under execution
4. After the execution of from clause select clause will execut.
5. In select clause we can pass three arguments
   (i) *
   (ii) column name
   (iii) Expression
6. The job of the select clause will reaches to the under execution table and searched the column name and display it in a output.
7. Therefor select clause is used to retrived the data from the table and display it.
8. The ~~opt~~ output of select clause is responsible for the result table or Output Table.

* Semicolon (;) : It is used to execut the query at the last line of a query we semicolon.

Note

SQL is not a case sensitive language.

Page No.
Date / /

Command's

☆ to set page alignments:

```
SET LINE 100 PAGES 100
```

☆ to clear Screen:

```
CL SCR OR CLEAR SCREEN
```

☆ to describe the table structure:

```
DESC TABLE_NAME;
```

Ex:- `DESC STUDENT;`

☆ Write a query to display all the Details from the database.

```
SELECT *
FROM TAB;
```

☆ Write the query to display Branches of ~~the~~ for the all the student.

⇒
```
SELECT branch
from student;
```

Page No.
Date

★ Write the query to display name and percentage for the all students.

⇒ select sname, per
From student;

★ Write a query to display Student id branch for all the students.

⇒ Select sid, Branch
from student;

Q. write a query to display Student id branch and Percentage for all the students.

⇒ Select sid, branch, per
from student;

Q. Write a query to display student id, student name, branch and Percentage for all the students.

⇒ select sid, sname, branch, per
from student;

✡ Astrics (*) :- It is used to display all details from the table.
Along with the (*) we can't use any column name or expression

Q. Write a query to display all the details from the students table.

⇒ select *
from student;

Page No.
Date / /

Q. Write a query to display all the details from the employee table

⇒
```
select *
from Emp;
```

Q. write a query to display all details from the department table:

⇒
```
select *
from DEPT;
```

Q. Write a query to display emp name and Hire Date for all the employees.

⇒
```
select ename, Hiredate
from Emp;
```

Q. Write a query to display salary, commission and dept department no for all the employe.

⇒
```
select sal, comm., dept_no
from emp;
```

Q. Write a query to display employee Number & job for all the employees.

⇒
```
select empno, job
from emp;
```

5 Rohan Soft

Page No.
Date / /

Q.ω

Q ① Write a query to display all the details from the employee table

⇒ select *
from EMP;

② WAQTD Name of All The employee:

⇒ select ENAME
from EMP;

③ WAQTD Name and salary given to all the employee.

⇒ select Ename, Sal
from Emp;

④ WAQTD Name and commission given to all the employees.

⇒ select ENAME, COMM
from EMP;

⑤ WAQTD Employee Id and Department Number of all the employee in Emp table.

⇒ select Empno ~~Empno~~, DEPNO
from EMP;

⑥ WAQTD ENAME And HIREDATE of All the Employee.

⇒ Select ENAME, Hiredate
from EMP;

⑦ WAQTD NAME AND Designation of All the employee.

⇒ Select ENAME, JOB
from EMP;

(8) WAQTD NAME, JOB and salary given all the employees.

⇒ Select ENAME, JOB, Sal
from Emp;

(9) WAQTD DNAME PRESENT IN Department table.

⇒ Select Dname
from Dept;

(10) WAQTD DNAME and location Present in Dept Table.

⇒ Select Dname, Loc
from Dept;

* Distinct clause: It is used to avoid duplicate or repeated value present in the result of a table

→ We can use distinct as a first argument in the Select clause.

→ In distinct we can use multiple columns. it will remove the combination of records which are get duplicate.

Q. Write a query to display different Sname present at the student table

⇒ Select Distinct Sname
from student;

Sid	branch	Per
1	MECH	90
2	CSE	80
3	EE	75
4	IT	90
5	CSE	80
6	CIVIL	65
7	MECH	80
8	BSC	70

No repetation

Note

select sname, distinct Per
from student;

→ We can not use distinct as a second argument.

Q. WAQTD different JOB present in a employee table.

⇒ select distinct JOB
from emp;

Q. WAQTD different JOB and salary present in a employee table.

⇒ select distinct job, sal
from emp;

Q. WAQTD different hiredate and dept. No. present in employee table

⇒ select distinct hiredate, deptno
from Emp;

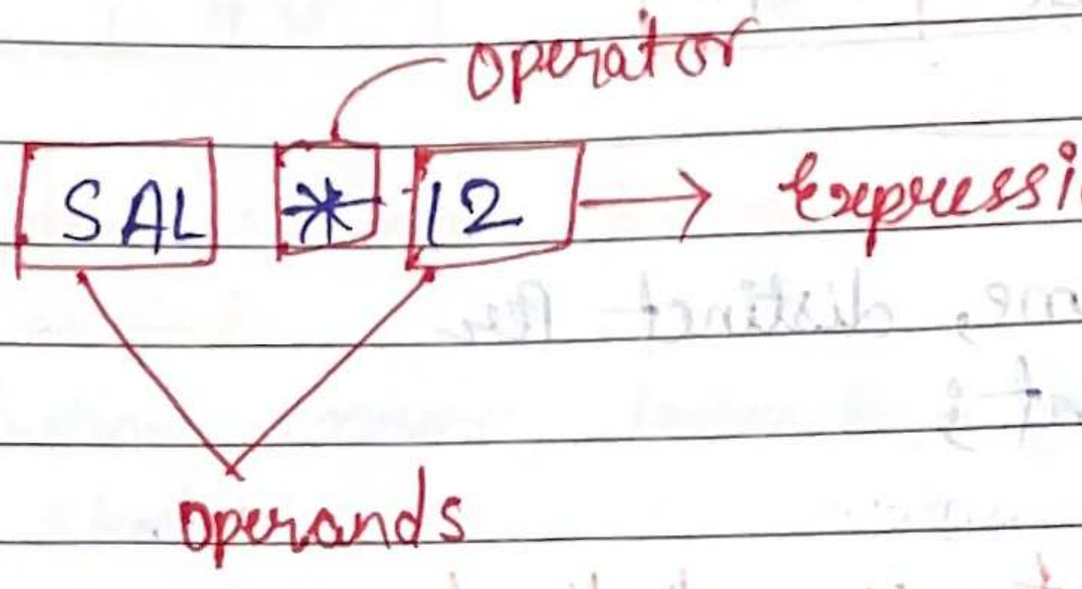
Page No.
Date / /

* **Expression :-** Any statement which gives result is known as expression.

5 + 5 = 10 Result
expression

Expression is always the combination of operends & operators.

Ex.



* **Operands :-** Operands may be a column name or values.

* **Operators :-** Operators are the predefine symbols used to perform a specific task.

Q. Write a query to display annual salary for all the employees.

⇒
```
Select sal*12
From emp;
```

Q. Write a query to display employee name half-term salary for all the employees.

⇒
```
select ename, sal*6
from emp;
```

Page No.
Date

Q. Write a query to display Salary and Salary with the hike of 500 for all the employees

⇒ select sal, sal+500
from emp;

Q. Write a query to display employee name, job, and salary with the deduction 300 for all the employees

⇒ Select ename, job, sal − 300
from emp;

Q. Write a query to display salary and Salary withe the hike of 10% for all the employees.

⇒ Select sal, ~~sal+10~~ sal + sal * 10/100
from emp;

Q. Note :- % with the hike (or) deduction

$$sal + sal * \frac{K}{100}$$

$$sal - sal * \frac{K}{100}$$

sal	sal + sal * 10/100
800	800 + 800 × 10/100
	= 800 + 80
	= 880.

Q. Write a query to display Annual Salary with the the deduction of 25% from the all employees

⇒ Select sal * 12 - sal * 12 * $\frac{25}{100}$
from emp;

Note :- % with like (or) deduction for annual salary

↳ sal * 12 + sal * 12 * $\frac{1\%}{100}$

↳ sal * 12 - sal * 12 * $\frac{1\%}{100}$

Q. Write a query to display all the details from the emp table along with annual salary for all the employees.

⇒ select emp.*, sal * 12
from emp;

Note : If we want to display all the details along with column name or expression we can use table name.*

Ex: emp.*

Q. Write a query to display Annual salary, Annual commision and salary with the deduction of 100 for all the employels.

⇒ Select sal * 12, comm * 12, sal - 100
from emp;

Q. Write a query to display annual salary with the hike of 10% and annual salary with the deduction of 30% for all the employees.

⇒ Select sal*12 + sal*12*10/100, sal*12 - sal*12*30/100 from emp;

Q. Write a query to display details of an employees along with annual commision for all the employees.

⇒ Select emp.*, comm*12 from emp;

* ALIAS :- Alias is a altrenative name give to a column name or expression in the result table.

↳ We can assign alias with or without a keyword (AS). That key

↳ Alias name will always a single string if it is multiple string we can use (" ") or under. (_)

Q. Write a query to display annual salary for all the employees.

⇒ Select sal*12 as Annual_sal from Emp;

possible ways →

sal*12 "Annual Salary"

sal*12 As Annual_salary

sal*12 Annual_salary

Sal*12 As "Annual_salary"

sal*12 "Annual_salary"

Q. Write a query to display salary and the salary with the deduction of 500 for all the employees.

⇒ Select sal, sal - 500 as "Salary deduction"
from Emp;

Q. Write a query to display employee name and salary with deduction of 25% for all the employees.

⇒ Select ename, sal - sal * 25/100 as "Deduction of 25%"
from Emp;

Q. Write name of the employee along with thier Annual salary.

⇒ Select ename, sal * 12 as "Annual Salary"
from emp;

Q. WAQTD ENAME and JOB for all the employee with their Half term Salary.

⇒ Select ename, job, sal * 6 as "Half term Salary"
from emp;

Q WAQTD All the details of the employee along with an Annual Bonus of 2000

⇒ Select Emp.*, sal * 12 + 2000 as "Annual Bonus"
from emp;

Q. WAQTD Name salary and salary with a Hike of 10%

⇒ Select ename, sal, sal + sal * 10/100 as "salary Hike 10%"
from emp;

Q. WAQTD Name and salary with deduction of 25%.

⇒ Select ename, sal - sal*25/100 as "salary deduction 25%"
from emp;

Q. WAQTD Name and salary with montly hike of 50

⇒ Select ename, sal + 50 as "Salary Montly Hike 50"
from emp;

Q. WAQTD NAME and Annual salary with deduction of 10%.

⇒ Select ename, sal*12 - sal*12*10/100 as "Annual salary"
from emp;

Q. WAQTD Total salary given to each employee (sal+comm)

⇒ Select sal + comm as "Total salary"
from emp;

Q.

Q WAQTD Details of all the employees along with annal salary.

⇒ Select emp.*, sal*12 as "Annual salary"
from emp;

Q. WAQTD Name and Designation along with 100 penalty in salary.

⇒ select ename, job, sal - 100 as "Penalty of salary"
from emp;

Q. Write a query to display details of an employees. along with annual comm for all the employees.

⇒ Select emp.*, sal*12 as "Annual Salary"
from emp;

Page No.
Date

Q. Write Quary to display Deptname & loc. present in department table.

⇒ Select dname, loc
from dept;

Q. WAQTD Different Job and Hiredate present at the employee table

⇒ Select job, hiredate
from emp;

Q. WAQTD Salary with the hike 13% for all the employees

⇒ Select sal * 13/100 As "Sal"
where emp;

Q. WAQTD Empname, Salary, commission for all the employess.

⇒ Select ename, sal, comm
from emp;

SELECTION

It is use to retrive data from the table by selecting both rows & columns.

Syntax:-

③ SELECT */[DISTICT] column_name/Expression [ALIAS]
① From TABLE_NAME
② WHERE <filter Condition>;

Order for execution

1 FROM
2 WHERE
3 SELECT

* WHERE clause :- where clause is used to filter the records (rows).

→ where clause execute after the from clause.
→ Where clause execut row by row.
→ In where clause we can use filter condition as argument.

column name / expression operator values/data

→ Where clause Boolen values. (true/false)
→ In where clause we can pass muttiple condition with the help of logical operators.

Q. Write a query to display ename & deptno. if the employees are working in deptement no. 20.

⇒ select Ename, deptno
from emp
where deptno = 20; → filter condition

Output of form clause

EMP

EmpNo	Ename	Sal	Deptno
1	ADAMS	1500	30
2	JONES	2750	10
3	ALLEN	850	30
4	MILLER	1200	20
5	JAMES	3000	10
6	WARD	2800	10
7	KING	1900	20

Under Execution

filter condition

Deptno=20

1	ADAMS	1500	30	(f)
2	JONES	2700	10	(f)
3	ALLEN	850	30	(f)
4	MILLER	1200	20	(T)
5	JAMES	3000	10	(f)
6	WARD	2800	10	(f)
7	KING	1900	20	(T)

Output of where clause

EmpNo	Ename	Sal	Deptno
4	MILLER	1200	20
7	KING	1900	20

⇩

Output of select clause

Ename	Deptno
MILLER	20
KING	20

Page No.
Date

Q. Write a query to display name, job and salary of an employees who are earn in salary less than 2000.

⇒ Select ename, job, sal
from emp
where sal < 2000;

Q. WAQTD name, job of employees who are working as a analyst.

⇒ select ename, job
from emp
Where job = ~~"analyst"~~ 'ANALYST';

Q. WAQTD details of an employees whoes name is ADAMS.

⇒ select *
from emp
Where ename = 'ADAMS';

Q. WAQTD details of the employees who are hire after 31-DEC-83.

⇒ select *
from emp
where hiredate > '31-DEC-83';

Q. WAQTD details of an employees along with annual salary if the employees annual salary is greater than 25000.

⇒ select emp.*, sal*12 as "Annual Salary"
from emp
where sal*12 > 25000;

Page No.
Date

Note we can not gives alias in where class because where class will execute before the select clause.

Q. WAQTD details of an employees who are hire after the year 82.

⇒ select emp.*
from emp
where hiredate > '31-DEC-82';

Note

* After the year 82 ---> 'DD-MON-YY'
After the last Date of year 82
After 31-DEC-82
> '31-DEC-82'

* Before the year 82
Before the first date of year 82
Before 01-JAN-82
< '01-JAN-82'

Q① WAQTD The annual salary of the employee whose name is smith.

→ Select sal*12 as "Annual salary"
from emp where ename = 'SMITH';

Q② WAQTD NAME of the employee working as clerk.

→ Select ename
from emp
where job = 'CLERK';

Q.3 Write a Salary of the employee who are working as SALESMAN

⇒
```
Select sal
from emp
where job = 'SALESMAN';
```

Q.4 WAQTD Details of the emp who earn more than 2000

⇒
```
Select *
from emp
where sal > 2000;
```

Q.5 WAQTD Details of the emp whose name is JONES

⇒
```
Select emp.*
from emp
where ename = 'JONES';
```

Q.6 WAQTD Details of the emp who was Hired After 01-JAN-81

⇒
```
Select emp.*
from emp
where Hiredate > '01-JAN-81';
```

Q.7 WAQTD Name & Sal along with His annual Salary If the annual salary is more than 12000.

⇒
```
select ename, sal*, sal*12 as "Annual Salary"
from emp
where sal*12 > 12000;
```

Q.8 WAQTD EMPNO of the Employee who are working in dept 30

⇒
```
select EmpNo
from emp
where Deptno = 30;
```

Q⑨ WAQTD ENAME and hiredate of they are hired Before 1981

⇒ Select ename, hiredate
from emp
where hiredate < '01-JAN-1981';

Q 10. WAQTD Details of the employee working as Manager

⇒ select emp.*
from emp
where job = 'MANAGER';

Q 11. WAQTD name and sal given to any employee if employee earn commision of Rupees 1400

⇒ select ename, sal
from emp
where comm >= 1400;

Q 12 WAQTD Details of employee Having commision more then salary.

⇒ Select emp.*
from emp
where comm > salary;

Q 13 WAQTD EMPNO of employee Having commision Hired Before The year 87.

⇒ Select empno
from emp
where Hiredate < '01-JAN-1987';

Q 14 WAQTD Details of employee working as an ANALYST.

⇒ select emp.*
from emp
where job = 'ANALYST';

Q15 WAQTD Details of Emps earning more than 20000 rupees per month.

⇒
```
select emp.*
from emp.
where sal >2000;
```

* OPERATORS IN SQL

1. Arithmetic operators: (+, −, *, /)
2. Concatenation Operators: (||)
3. Comparision operators: (=, != , <>)
4. Relational operators: (>, <, >=, <=)
5. logical operator: (AND, OR, NOT)
6. special Operator: (1. IN
   2. NOT IN
   3. BETWEEN
   4. NOT BETWEEN
   5. IS
   6. IS NOT
   7. LIKE
   8. NOT LIKE)
7. SUBQUERY OPERATORS: (1. ALL
   2. ANY
   3. EXISTS
   4. NOT EXISTS)

2. Concatenation operator (||): It is used to join the given strings.

→ this symbol for a concatenation operator "||".

Page No.
Date

Ex:-

1. Select 'MR.' ||ENAME
From Emp;

'MR.' ||ENAME

MR. SMITH
MR. ALLEN

2. Select 'HII' ||ENAME|| 'UR SALARY IS' ||SAL
From EMP;

'HII' ||ENAME||' UR SALARY IS' ||SAL
HII SMITH UR SALARY IS 800

3. SELECT 'HII' ||ENAME||'UR GF NAME IS MIA'
FROM EMP
WHERE ENAME = 'SMITH';

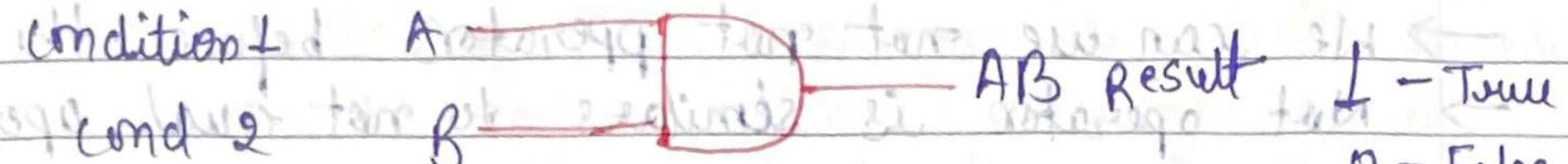
'HII' ||ENAME||' UR GF NAME IS MIA'

HII SMITH UR GF NAME IS MIA

* Logical Operators:- It is used to write multiple conditions in a where clause.
(AND, OR, NOT)

* AND:- And operator is a Binary Multiplication operator.

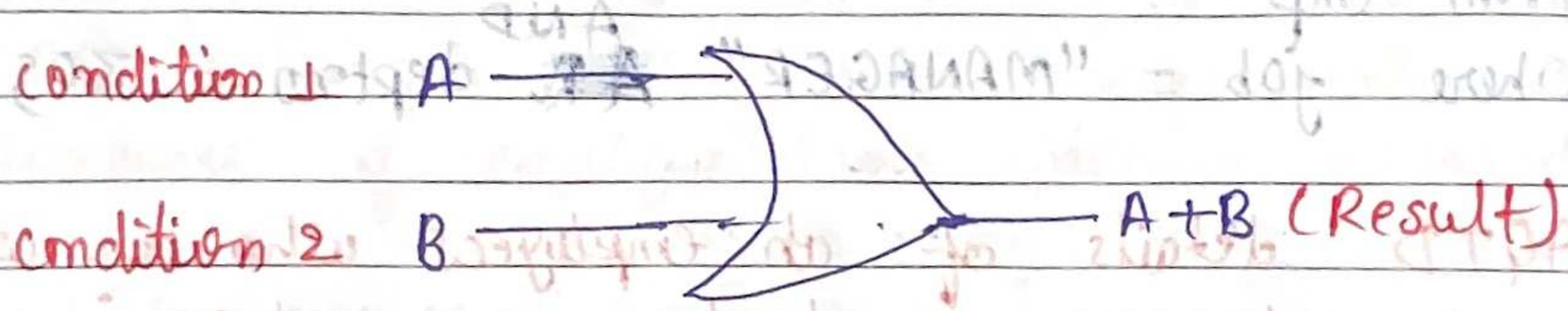
→ And operator returns true if all the conditions get satisfied.



$2^2 = 4$

C_1 A	C_2 B	Result AB
T	F	F
F	T	F
F	F	F
T	T	T

* OR : OR operator is a Binary addition operator
→ OR operator returns true if any one of the condition get satisfied.



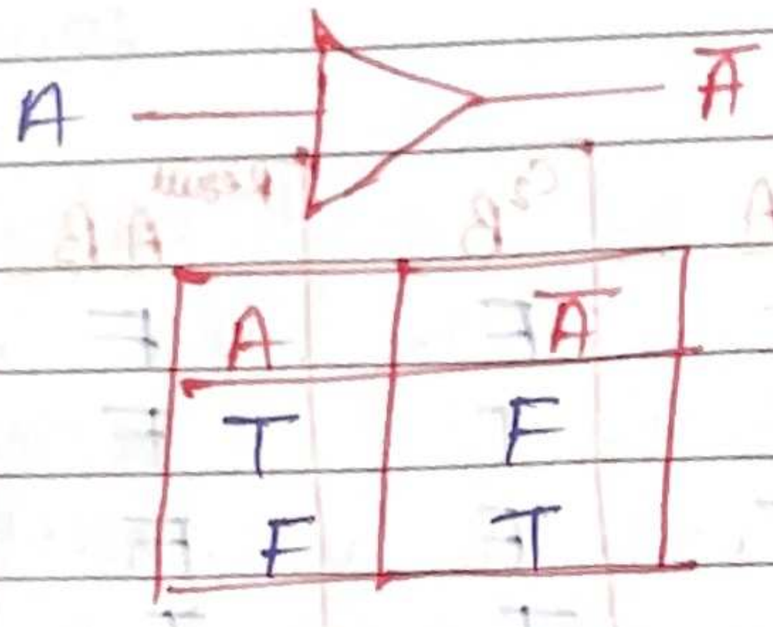
$2^2 = 4$

C_1 A	C_2 B	Result A+B
T	F	T
F	T	T
T	T	T
F	F	F

* NOT : NOT operator is a inverse operator and is also known as unary operator. (It can accept only one operand) at a right hand side.

→ Not operator is used to inverse the results.

→ We can use not not operators before the condition.
→ Not operator is similars to not equal operator.



A	Ā
T	F
F	T

Ex.
Deptno != 20
Deptno <>20
NOT Deptno =20

Q.1. WAQTD Job and departement no. of an employee, if the employess are working as MANAGER in departement no. 30

⇒ select Job, deptno
from emp
where job = "MANAGER" AND deptno = 30;

Q.2. WAQTD details of an employees who are working as a salsemane in deptno 30 and earning salary greater than 1000.

⇒ select *
from emp
where job = "SALESMAN" AND Deptno = 30
AND sal > 1000;

Q 3 WAQTD name and salary of an employees who are earning salary greater then 1000 and less than 3000.

⇒ select ename, sal
from emp
where sal >1000 AND sal < 3000;

Page No.
Date

Q4 WAQTD details of an employees along with annual salary if the employees are working as a ANALYST and hire before the year 82.

=> Select * emp.*, sal*12 as "ANNUAL-SAL"
from emp
Where job = 'ANALYST' AND hiredate < '01-JAN-82';

Q5 WAQTD name and job of the employee if the employees are working as a MANAGER OR a SALESMAN.

=> Select ename, job
from emp
where job = 'MANAGER' OR job = 'SALESMAN';

Q6 WAQTD name salary and department no. of employess if employees are working of a dept of 10, 20, 40

=> Select ename, sal, deptno
from emp
Where deptno = 10 OR deptno = 20 OR deptno = 40;

Q7 WAQTD details of an employee's except who are working in dept no 30.

=> select *
from emp
where NOT deptno 30;
or "deptno != 30;
or deptno <> 30;

Q.8 Write TD name and salary of an employees except who are earning Salary less than 1000.

⇒ select ename, sal
from emp
where NOT sal < 1000;
or sal >= 1000;

Q.9. WAQTD details of an employees working as MANAGER in dept no. 10 OR 20.

⇒ select *
from emp
where job = 'MANAGER' AND (deptno = 10 OR deptno = 20);

Q 10 WAQTD name, job and deptno of employees if the employees are working as a clerk or MANAGER in Dept 20

select ename, job, deptno
from emp
where (job = 'CLERK' OR job = 'MANAGER')
AND deptno = 20;

Q 11 WAQTD details of an employees if the employees are working as clerk or ANALYST in dept No. 10, OR 20

⇒ select *
from emp
where (job = "CLERK" OR job = 'ANALYST')
AND (deptno = 10 OR deptno = 20);

Q 12 WAQTD name and deptno of an employes except who are working in deptno 10 OR 20.

⇒ select ename, deptno
from emp
where deptno != 10 AND deptno != 20;
OR NOT (deptno = 10 OR deptno = 20);

Q13 WAQTD details of employees exept who are working as a SALESMAN OR ANALYST

⇒ select *
from emp
where NOT (job = 'SALESMAN' OR job = 'ANALYST'
or job != 'SALESMAN' AND job != 'ANALYST'

Assignment

Q.1 WAQTD Details of the employee working as clerk and earning less than 1500

⇒ select *
from emp
Where job = 'CLERK' AND Salary < 1500;

Q2 WAQTD Name & Hiredate of the employees working as MANAGER IN Dept 30

⇒ Select ename, hiredate
from emp
where job = 'MANAGER' AND Deptno = 30;

Q3 WAQTD details of employees Along with annual salary if they are working in dept 30 as SALSEMAN and Their Annual salary has to be greater than 14000

⇒ Select emp.*, sal*12 AS "Annual Salary"
from emp
where deptno = 30 AND job = 'SALESMAN' AND
Sal*12 > 14000;

Q4 WAQTD ALL The details of the emp working in dept 30 OR AS ANALYST

⇒ select *
from emp
where deptno = 30 OR job = 'ANALYST';

Q5 WAQTD Name of the employees whose salary is less than 1100 and their dessignation is clerk.

⇒ select ename
from emp
where sal < 1100 AND job = 'CLERK';

Q6 WAQTD name and sal, Annual sal and deptno If Deptno is 20 Earning more than 1100 and annual sal exceeds 12000

⇒ select ename, sal, sal*12 As "Annual Salary", deptno
from emp
where deptno = 20 AND sal > 1100 AND sal*12 > 12000;

Q7 WAQTD Empno and Name of the employees working as manager in dept no 20.

⇒ Select empno, ename
from emp
where job = 'MANAGER' AND deptno = 20;

Q8 WAQTD details of employees working in dept 20 OR 30

⇒ select *
from emp
where deptno = 20 ORA deptno = 30;

Page No.
Date

Q 9 WAQTD details of employees working as Analyst in dept 10

⇒ select *
from emp
where job = 'ANALYST' AND deptno = 10;

Q 10 WAQTD details of employee working as PRESIDENT with Salary of Rupees 4000.

⇒ select *
from emp
where job = 'PRESIDENT' AND sal = 4000;

Q 11 WAQTD NAME and deptno, job of emps working as clerk in dept 10 or 20.

⇒ select ename, deptno, job
from emp
where job = 'CLERK' AND (deptno = 10 OR deptno = 20);

Q 12 WAQTD details of employee working as clerk or Manager in dept 10.

⇒ select *
from emp
where (job = 'CLERK' OR job = 'MANAGER')
AND deptno = 10;

Q 13 WAQTD Name of employees working in Dept 10, 20, 30, 40

⇒ select ename
from emp
where deptno = 10 OR deptno = 20 OR deptno = 30
OR deptno = 40;

Q 14 WAQTD details of employee who with empno 7902, 7839

⇒ select *
from emp
where empno = 7902 AND empno = 7839;

Q 15 WAQTD details of employee working as MANGER OR SALESMAN OR CLERK

⇒ select *
from emp
where job = 'MANAGER' OR job = 'SALESMAN' OR job = "CLERK";

Q 16 WAQTD name of employee hiredate after 81 and Before 87

⇒ select ename
from emp
where hiredate > '31-DEC-81' AND hiredate < '01-JAN-82';

Q 17 WAQTD details of employee earning more than 1250 but less than 3000

⇒ select *
from emp
where sal > 1250 AND sal < 3000;

Q 18 WAQTD name of employees hire after 81 into dept 10 OR 30

⇒ select ename
from emp
where hiredate > '31-DEC-81' AND (deptno = 10 OR deptno = 30);

Page No.
Date

Q 19 WAQTD name of employees along with annual salary for the employee working as manager or clerk into dept 10 or 30.

⇒
```
select ename, sal*12 AS "Annual salary"
from emp
  where ( job = 'MANAGER' OR job = 'CLERK') AND
        (deptno =10 OR deptno =30);
```

Q 20 WAQTD All the details along with annual salary if sal is between 1000 AND 4000 Annual salary more than 15000

⇒
```
select *emp.*, sal*12, AS "Annual salary"
from emp
where SAL>1000 AND Sal<4000 AND Sal*12>15000;
```

Q 21 WAQTD details of all the employee ~~except~~ except the emps working is dept 10 OR 30.

⇒
```
select *
from emp
where NOT (deptno =10 OR deptno =30);
```

Q 22 WAQTD details of all emps along with annual salary except the employees working as ANALYST OR MANAGER.
```
select emp.*, sal*12 AS "Annual salary"
from emp
where NOT (job = 'ANALYST' OR job ='MANAGER');
```

Page No.

Date

(9) 10 IN (10, 20, 30) T

(8) 40 IN (10, 20, 30) F

an employees. working in department
clerk.

MANAGER, CLERK

Page No.
Date

* Special Operator :-

1. IN
2. NOT IN
3. BETWEEN
4. NOT BETWEEN
5. IS
6. ISNOT
7. LIKE
8. NOT LIKE

1. **IN OPERATOR :-** In operator is similare to equals operator.

→ In operator is also known as multivalued Operator.

→ In operator can accept multiple values. belongs to a same column in a Right hand side (RHS)

→ In operator returns true if any one of the condition get satisfied.

```
ColumnName/Expression IN (V1, V2, V3 - - - - Vn)
```

Q. 1. HAVE names of an employees who are working in the dept of 10, 20, 30.

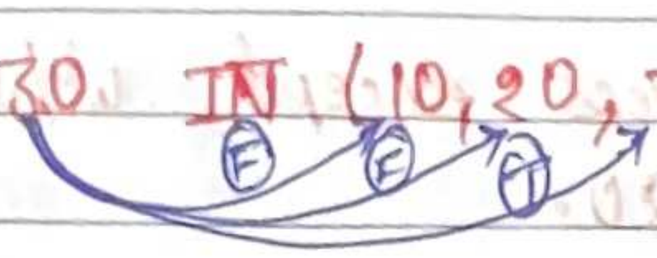
⇒ emp

Ename	sal	Deptno
ALLEN	1800	30
BLAKE	2500	10
JONES	3000	20
ADAMS	2500	40
KING	1950	10

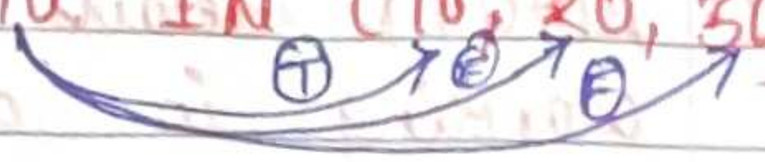
```
Select Ename
from emp
Where deptno IN (10,20,30)
```

30	T
10	T
20	T
40	F
10	T

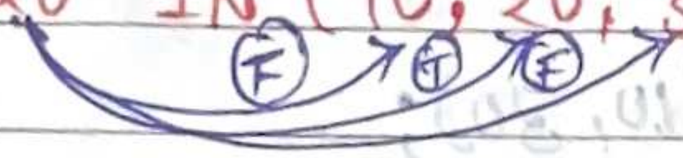
① 30 IN (10,20,30) T



② 10 IN (10,20,30) T



③ 20 IN (10,20,30) T



④ 40 IN (10,20,30) F

⑤ 10 IN (10,20,30) T

Q.2 WAQTD details of an employees working in deptno 10 as Manager or clerk.

⇒ select *
from emp
where Deptno = 10 AND job IN ('MANAGER', 'CLERK');

Q3. WAQTD name, job and deptno of an employees if the employees are working as CLERK or ANALYST is dept NO 10 or 20.

⇒ Select ename, job, deptno
from emp
where job IN ('CLERK', 'ANALYST') AND Deptno IN (10,20);

2. NOT IN : It is similare to IN operator instead of selecting the value it rejects the value.

Syntax

ColumnName/Expression NOT IN (V_1, V_2 ... V_n)

Q.1: WAQTD details of an employees except who are working in deptno 10, 20.

⇒ select *
from emp
where deptno NOT IN (10, 30);

Q.2 WAQTD details of employees along with annual salary if the employees are working in deptno 10 OR 20. except who are MANAGER OR ANALYST.

⇒ select emp.*, sal * 12 AS "Annual Salary"
from emp
where deptno IN (10, 20) AND job NOT IN ('MANAGER', 'ANALYST');

3. IS operator : It is used to compare with only the null value.

Syntax :

column_name/Expression IS Null;

Q1. WAQTD details of an employees if the employees are not earning any commission.

⇒ select *
from Emp
where comm IS NULL;
or Icomn IS NULL OR comm = 0;

Q2. WAQTD name and Manager no (MGR) of the employees if the does not have any repeating as MANAGER. (does have a Manager NO)

⇒ select ename, MGR

from emp
where MGR IS NULL;

4. IS NOT : It is used to compare with Not null values.

Syntax
column_name/expression IS NOT NULL;

Q1. WAQTD details of an employees if the employees are earning some commision.

⇒ select *
from emp
Where comm IS NOT NULL;

Q2. WAQTD Name, salary and commision of the employee if the employees are earning some salary but not a commission.

⇒ select ename, sal, comm
~~where~~ from emp
where sal IS NOT Null AND Comm IS Null;

5. Between : [Range of values] :

→ Whenever we have range of values we go for between operater. (Starting value and Ending value).
→ Between operator works including the ranges.
→ We can not interchange rang of values in between operator.

ColumnName/Expression Between Lower Range AND Higher Range

Q.1 WAQTD Name of an employees if the employees are earning salary greater than 1500 and less than 3000.

⇒ Range : 1501 AND 2999

```
select ename
From emp
where sal Between 1501 AND 2999;
```

(sal: Column Name; 1501: Lower Range; 2999: Higher Range)

emp

Ename	sal	Deptno
JONES	1900	10
ALLEN	1500	30
ADAM	3500	20
KING	800	10
MILLER	4000	40

1950 Between 1501 AND 2999 → >= (T) <= (T)

1500 Between 1501 AND 2999; → >= (F) <= (T)

3500 Between 1501 AND 2999; → >= (T) <= (F)

Note

800 Between 1501 AND 2999 → >= F <= T

4000 Between 1501 AND 2999 → >= T <= F

Note

Sal > 1500 And Sal < 3000 → Range : 1501 And 2999

Salary Ranges from 1500 to 3000 → Range : 1500 And 3000

Salary Between 1500 to 3000 → Range : 1501 And 2999

Page No.
Date

Q.1 WAQTD details of an employees if the employees are earning salary in the range of 2000 to 3000.

⇒
```
select *
from emp
where sal Between 2000 AND 3000;
```

Q.2 WAQTD name and hiredate of employees if the emp are hired during the year 81

⇒
```
select ename, hiredate
from emp
where hiredate Between '01-JAN-81' AND '31-DEC-81';
```

Q.3 WAQT details of an employees who are hire after the year 81 but before the year 84.

⇒
```
Select *
from emp
where hiredate Between '01-JAN-82' AND '31-DEC-83';
```

8. NOT BETWEEN :-

Not Between operater is similar to between operator but it will reject the range of value.

Syntax

columnName/Expression NOT BETWEEN lower Range AND Higher Range

Q. WAQTD details of an employees except who are earning salary in the ~~rng~~ range of 2500 to 5000.

⇒
```
select *
from emp
where sal NOT BETWEEN 2500 AND 5000;
```

Q 2 WAQTD hiredates of an employees except who are hire during the year 81

⇒ select hiredate
from emp
where hiredate NOT BETWEEN '01-JAN-81' AND '31-DEC-81';

Q 3 WAQTD name and salary of an employees who are earning salary less than 1000 and greater than 3000.

⇒ select ename, sal
from emp
where sal NOT Between 1000 AND 3000;

7. LIKE : It is used to Matching the pattern.

Syntax

ColumnName / Expression LIKE "pattern to Match";

1. % matching the pattern in LIKE Operator we use two special characters.

1. % —→
2. _ (underscore)

1. % :- one % can accept 'n' numbers of characters, any characters but some times no characters.

(# if we know the length of character)

2 _ (Underscore) :- One '_' can accept ~~four~~ ~~accept~~ exactly only one character but any character.

Q1 WAQTD names of an employees if the employees name starts withe character 'A'?

⇒ select ename
from emp
where ename LIKE 'A%';

Q 2 WAQTD names of an employees if the employees names end with character 'S';
select ename
from emp
where ename LIKE '%S';

Q 3 WAQTD names of employees if the employee name satarts with 'J' and end with chracter 'S'.

⇒ select ename
from emp
where ename LIKE 'J% S';

Q4 WAQTD name of the employees if the employees name starts with 'S'.

⇒ select ename
from emp
where ename LIke 'S%';

Q 5. WAQTD Names of an employees if the employee name has charater 'A'.

⇒ Select ename
from emp
where ename like '%A%';

Page No.
Date

for

WARD BLAKE ALLEN ADAMS
%A% %A% %A% %A%

Q.6 WAQTD name, salary, hiredate of the emp. If the emp hire during the year 81.

⇒ select ename, salary, hiredate
from emp
where hiredate LIKE '%81';
'--------81';

Q7 WAQTD details of an employees if the employee are hired in the month of dec;

⇒ select *
from emp
where hiredate LIKE '%DEC%';
'---DEC---';

Q 8. WAQTD name of an employees if the employees name has exactly 4 character.

⇒ select ename
from emp
where ename LIKE '----';

* Q 9. WAQTD details of an employee if the employee name has character 'A' in as a the second caracter

⇒ select *
from emp
where ename LIKE '_A%';

Page No.
Date / /

Q.10 WAQTD name of the employees if the employee name has character 'L' as 2nd charater char 'R' as a last char.

⇒
```
select ename
from emp
where ENAME LIKE '_L%R';
```

Q.11 WAQTD details of employees if the emp. name has 3rd last char as a 'E'.

⇒
```
select *
from emp
where ename LIKE '%E__';
```

Q.12 WAQTD name of an employees if the employee name has char 'A' has occured for Twice.

⇒
```
Select ename
from emp
where ename LIKE '%A%A%';
```

Q.13 NOT LIKE OPERATOR

not liake operator is similer to like operator it will rejects the pattern.

Syntax

```
ColumnName/Expression NOT LIKE 'Patter to match'
```

Q.1. WAQTD employee name except who name has char 'A'.

⇒
```
select ename
from emp
where ename NOT LIKE '%A%';
```

Page No.
Date

Q 2. WAQTD details of an employees except who are hire int the month of FEB.

⇒
```
Select *
from emp
where hiredate NOT LIKE '%FEB%';
```

Q 3 WAQTD name, job & Salary of emp except who are earning 4 digit Salary.

⇒
```
select ename, job, sal
from emp
where sal NOT LIKE '____';
```

Q 4 WAQTD Employee name if the employee name has char 'A' has occured for exact one Time.

⇒
```
Select ename
from emp
where ename LIKE '%A%' AND
      ename NOT LIKE '%A%A%';
```

name display

Q 5 WAQTD ^name start with consonants.

⇒
```
Select ename
from emp
where ename NOT LIKE 'A%' AND ename NOT LIKE
'E%' AND ename NOT LIKE 'I%'
AND ename NOT LIKE 'O%'
AND ename NOT LIKE 'U%';
```

1. WAQTD Details of the employee who's name start with 'A'.

⇒ Select *
from emp
where ename LIKE 'A%';

2. WAQTD Details of the employee who's Name End with 'S'.

⇒ Select *
from emp
where ename LIKE '%S';

3. WAQTD Details of the employee who's name starts with 'A' and END with 'S'.

⇒ select *
from emp
where ename LIKE 'A%S';

4. WAQTD Details of the employee who's name starts with 'A' AND END with 'A'

⇒ Select *
from emp
where ename LIKE 'A%A';

5. WAQTD Details of employee who's name ~~starts with 'A'~~ has charcter 'R' In It.

⇒ Select *
from emp
where ename LIKE '%R%';

Page No.
Date

Q 6 WAQTD details of the employee who's name has character 'L' Present as the 3rd character.

⇒
```
select *
from emp
where ename LIKE '__L%';
```

Q 7 WAQTD details of the employee who's name has character 'L' present as the 3rd and 4th character.

⇒
```
select *
from emp
where ename LIKE '__LL%';
```

Q 8 WAQTD Details of the employee who's name has character 'A' as the 1st and character 'M' as the Last But one IN The Name (Last 2nd char)

⇒
```
Select *
from emp LIKE
where ename^'A%M';
```

Q 9 WAQTD details of the employee who's name has Excatly 4 character.

⇒
```
Select *
from emp
where ename LIKE '____';
```

Q 10 WAQTD ENAME AND hiredate of the employee who where hired during The year 82.

⇒
```
select ename, hiredate
from emp
where hiredate LIKE '%82';
```

11. WAQTD ENAME and hiredate of the employee who were hired is the month of FEB.

⇒ select ename, hiredate
from emp
where hiredate LIKE '%FEB%';

12. WAQTD ENAME and Sal of employee who are earning a salary of 3 digit.

⇒ select ename, sal
from emp
where sal LIKE '---';

13. WAQTD ename, hiredate and sal of the employee who are earning a salary of 3 digit And where hired In the month of DEC.

⇒ select ename, hiredate, sal
from emp
where sal LIKE '---' AND hiredate LIKE '%DEC%';

14 WAQTD ename and sal of employee who are earning a sal of 4 digit and their name start with 'J'.

⇒ Select ename, sal
from emp
where sal LIKE '----' AND ename LIKE 'J%';

15 WAQTD ename of the employee who's name has 5 char. in which the 4th character is 'E'.

⇒ select ename
from emp
where ename LIKE '___E_';

Q1 WAQTD name of an employe if the employees name start with @.

⇒ Select ename
from emp
where ename LIKE '@%';

Q2 WAQTD ename of if the name starts with %.

⇒ select ename
from ~~emp~~ emp
where ename LIKE '!%%' Escape '!';

ESCAPE CHARACTER:

It is used to remove the special behaviour given to a special character (%, _) and to treat it as a normal character.

Syntax:
column name/expression LIKE/NOT LIKE 'pattern to match' ESCAPE 'char';

Note

1. Escape character must be defined
2. Escape character must be used before the special character which has to treated as a normal character.
3. The recommended characters for escape is:
   [!, /, \, $]

Page No.
Date

Q. select ename
From emp
where Ename like '!%%' escape '!';

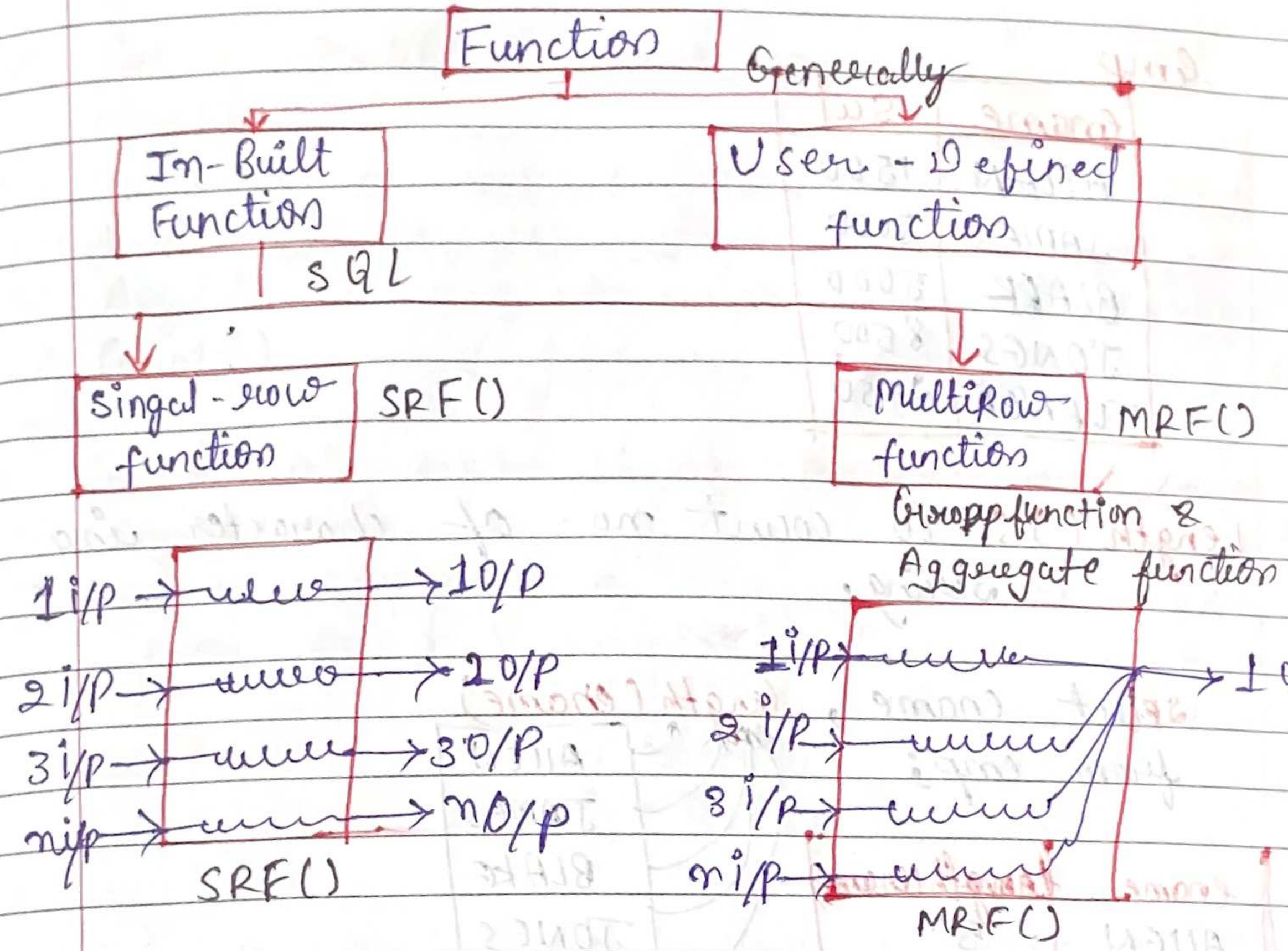
Q1 WAQTD ename if the name end with underscore
⇒ select ename
from emp
where ename like '%!_' escape '!';

Q2 WAQTD name of an employees if the employees name with % and end with (—).
⇒ select ename
from emp
where ename like '%!%!_' Escape '!';
'!%%!_'

Q3 WAQTD employees name if the employees name has underscore (—)
⇒ select ename
from emp
where ename like '%!_%!_' Escape '!';
'%!_%'

Q4. WAQTD if the employee name has %.
⇒ select ename
from emp
where ename like '%!%%' escape '!';

Functions: functions are the list of codes or block of instruction used to perform a specific task.



1. Singal row function Single row function execute row by row

→ In Single row function if we pass 1 input is process execute and generate only one output.

→ If we pass 'n' number of I/P it generate 'n' number of O/P.

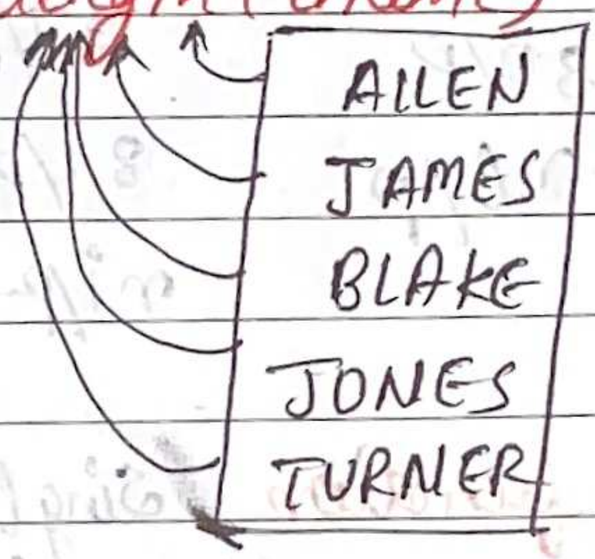
Q. Write a query to display number of characters present in employee name along with employee name for all the employees.

emp

ename	sal
ALLEN	1500
JAMES	3000
BLAKE	5000
JONES	8500
TURNER	2750

* Length(): To count no. of character in a string.

```
select ename, length(ename)
from emp;
```



ename	length(ename)
ALLEN	5
JAMES	5
BLAKE	5
JONES	5
TURNER	6

2. Multi row function / group function / Aggregate functions: multi row function Aggregate all the records at the single short function and then it will executes.

→ In multirow function if we pass one input it

generates one output.

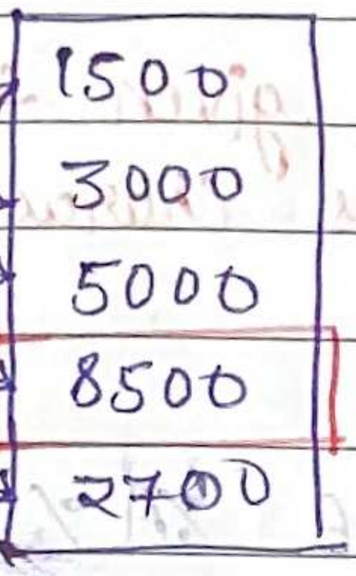
→ In multi-row function if we pass 'n' number of input it generates exactly only one output.

* List of multi-row function

1) Max()
2) Min()
3) Sum()
4) Avg()
5) Count()

Q1 write a query to display maximum salary from the employee table.

⇒ select Max(sal) from emp;



output:

Max(sal)
8500

* Rules of Multi-row function:

→ In multi row function we can use single ~~eg~~ arguments (i/p) it may be column name or expression.

→ along with a multirow function we can not use column name or expression.

→ We can not use multi row function in where clause.

→ Multi row function ignores null values.

→ Count() is the only multirow function which accept astriks (*) as a argument.

Q 1. WAQTD average salary given to the employees if the employees are working as a salesman.

⇒
```
select Avg(sal)
from emp
where job = 'SALESMAN';
```

Q 2 WAQTD maximum and minimum salary given to the employees if the employees are working in dept no (10, 20). and hired before the year 82.

⇒
```
select Max(sal), Min(sal)
from emp
where deptno IN (10, 20) AND hiredate < '01-JAN-82';
```

Q 3 WAQTD total salary given to the employees if the employee name has character 'A'.

⇒
```
select sum(sal)
from emp
where ename LIKE '%A%';
```

Q 4 WAQTD number of employees working as a salesman Or Manager.

⇒
```
select count(*)
from emp
where job IN ('SALESMAN', 'MANAGER');
```

Q 5 WAQTD total salary and number of employees if the employees are working in deptno 30 and earning some commision.

⇒
```
select sum(sal), count(*)
from emp
where deptno = 30 AND comm IS NOT Null;
```

Page No.
Date

Q6: WAQTD no. of distinct job present in employee table.
⇒ Select ~~Distinct~~ ~~count(job)~~ count(Distinct job);
from emp;

Q1 WAQTD number of employees getting salary less then 2000 In dept no 10.
⇒ select ~~empno~~ count(*)
from emp
where sal < 2000 AND deptno = 10;

Q2 WAQTD Total salary needed to pay employees working as clerk.
⇒ select sum(sal)
from emp
where job = 'CLERK';

Q3 WAQTD Average salary needed to pay all employee
⇒ select avg(sal)
from emp;

Q4 WAQTD number of employee having 'A' as their first character.
⇒ select count(*)
from emp
where ename LIKE '%A%';

Q5 WAQTD number of employee working as clerk or manager.
⇒ select count(*)
from emp
where job IN ('CLERK', 'MANAGER');

Page No.
Date

Q6 WAQTD Total salary needed to pay employees hired in FEB.

⇒ Select sum(sal)
from emp
where hiredate LIKE '%FEB%';

Q7 WAQTD number of employees reporting to 7839 (MGR)

⇒ select count(*)
from emp
where MGR = 7839;

Q8. WAQTD number of employees getting commision in deptno 30

⇒ select count(*)
from emp
where comm IS NOT NULL AND deptno = 30;

Q9 WAQTD avg sal, total sal, number of emps and maximum salary given to employees working as president.

⇒ Select avg(sal), Sum(sal), count(*), max(sal)
from emp
where job = 'PRESIDENT';

Q10 WAQTD number of employees having 'A' In their names

⇒ select count(*)
from emp
where ename LIKE '%A%';

Page No.
Date

Q11 WAQTD number of emps and total salary needed to pay the employees who have 2 consicutive L's In their names.

⇒ select count(*), sum(sal)
from emp
where ename like '%LL%';

Q12 WAQTD number of distincy departments present in employee table.

⇒ select count(distinct deptno)
from emp;
~~where~~

Q13 WAQTD number of employees having character '_' in their name.

⇒ select count(*)
from emp
where ename LIKE '%!_%' escape '!';

Q14 WAQTD number of employees having atleast 2 Percentles (%) in their their names.

⇒ select count(*)
from emp
where ename like '%!%%!%%' Escap '!';

Q15 WAQTD Total salary given to employees working as cleark in dept 30.

⇒ select sum(sal)
from emp
where job = 'CLERK' AND deptno = 30;

Q16 WAQTD maximum Salary given to the employees working as analyst.

⇒ select max(sal)
from emp
where job = 'ANALYST';

Q17 WAQTD number of distinct salarys present in employee Table.

⇒ select count (distinct sal)
from emp;

Q18 WAQTD number of different jobs present in employee table.

⇒ select count (distinct job)
from emp;

Q19 WAQTD avg Salary given to the clerk.

⇒ select avg(sal)
from emp
where job = 'CLERK';

Q20 WAQTD minimum Salary given to the employees who work in dept 10 as manger or a clerk.

⇒ select min(sal)
from emp
where deptno = 10 AND job IN ('MANAGER',
'CLERK');

Grouping and Filter:

1. Group by clause: It is used to grop group the records in each wish there records.

Syntax

SELECT group by expression / group_function
from tablename
[WHERE <filter_condition>] -
Group By columnName / expression;

ORDER of execution

1 FROM
2 WHERE (if used) [ROW - BY - ROW]
3 Group By [ROW - BY - ROW]
4 SELECT [GROUP - BY - GROUP]

* Group By clause - It is used to group the records.
→ group by clause executes row - by - row.
→ In group by clause we can pass columnname or expression as a ~~ag~~ argument (
→ After the execution of group by clause we are getting groups.
→ Therefore any clause execute after the group by it will executes group - by - group.
→ without where clause also we can use group by clause.

Page No.
Date

Note:

group by expression:

1. The columnname has to be used in group by clause the same column we can access in select clause with the multi-row function also. those column are known as group by expression.

Q1 WAQTD number of employees working in each dept. (multi row function)

⇒ Group function

```
Select count(*), Deptno → Group by expression
from emp
Group by deptno;
        column name
```

EMP

Emp

Database → emp

EMP

Empno	Ename	Sal	Deptno
1	ADAMS	1250	30
2	JAMES	3000	10
3	BLAKE	1850	10
4	JONES	3500	20
5	MILLER	1900	10
6	Turner	2750	30
7	KING	1850	20

Output of from clause

Output of group by clause

30

1	ADAMS	1250	30
6	TURNER	2750	30

10

2	JAMES	3000	10
3	BLAKE	1850	10
5	MILLER	1900	10

20

4	JONES	3500	20
7	KING	1850	20

count(*)

30 → 2

10 → 3

20 → 2

Output of select clause

count(*)	Deptno
2	30
3	10
2	20

Page No.
Date

Q 2 WAQTD Maximum salary given to the employees in each job if the emps are hired during the year 81.

⇒ Select max(sal)
from emp
where hiredate between '01-JAN-81' AND '31-DEC-81'
group by job;

Q 3 WAQTD Avg salary given to employees in job wise.

⇒ Select Avg(sal), job
from emp
group by job;

Q 4 WAQTD Total salary and number of employees in each dept if the employees are working as Manager or salesman.

⇒ select sum(sal), count(*), deptno
from emp
where job IN ('MANAGER', 'SALESMAN')
grob by deptno;

Q 5. WAQTD maximum and minimum salary given to employees in department wise

⇒ select max(sal), min(sal), deptno
from emp
group by deptno;

Q 6. WAQTD number of employees working in each department except who are working as analyst.

⇒ select count(*), deptno
from emp
where job NOT IN 'ANALYST'

group by deptno;

Q1 WAQTD Number of employes working in each departmen except president.

⇒ select count(*), deptno
from emp
where job not IN 'PRESIDENT'
group by deptno;

Q2 WAQTD Total salary needed to pay all the employees in each job.

⇒ select sum(sal), job
from emp
group by job;

Q3 WAQTD number of employees working as manager in each Departement.

⇒ select count(*), deptno
from emp
where job = 'MANAGER'
group by deptno;

Q4. WAQTD ~~number of em~~ avg salary needed to pay all the employees in each department excluding the employees of deptno 20

⇒ select avg(sal), deptno
from emp
where deptno = 20
group by deptno;

Q5 WAQTD Number of employes having character 'A' In their name in each job.

⇒ select count(*), job
from emp
where ename LIKE '%A%'
group by ~~ename~~ job;

Page No.
Date / /

Q6 WAQTD number of employees and avg salary needed to pay the employees who salary in greater then 2000 in each dept.

⇒ select count(*), avg(sal), deptno
from emp
where sal > 2000
group by deptno;

Q7 WAQTD Total salary needed to pay and number of salasmans in each Dept:-

⇒ select sum(sal), count(*), deptno
from emp
where job = 'SALESMAN'
group by deptno;

Q8 WAQTD number of employees what with their maximum salary in each job.

⇒ Select count(*), max(sal), job
from emp
group by job;

Q9 WAQTD maximum salary given to an employee working in each Dept:-

⇒ select max(sal), deptno
from emp
~~where~~ group by deptno;

Page No.
Date

Q10 WAQTD number of times the salaries have been repeated in employee table.

→ select count(sal), sal
from emp
group by sal;

Having clause

→ It is used to filter the groups. with the help of condition

Syntax

```
Select group_by_expression/group_function
From Table_name
[where <filter_condition>]
GROUP By column_name/expression
Having <group_filter_condition>;
```

→ Order of execution:

1. From
2. WHERE (if used) [Row-By-Row]
3. Group By [Row-By-Row]
4. HAVING [Group-By-Group]
5. SELECT [Group-By-Group]

Database

output of from clause

Emp

Emp no	Ename	Sal	Deptno	
1	Scott	1800	30	(T)
2	Jones	1950	20	(T)
3	James	1200	20	(F)
4	Adams	3000	10	(T)
5	Blake	1500	10	(F)
6	Smith	2750	30	(T)
7	Miller	2000	20	(T)
8	King	3500	30	(T)

Under execution

Output of where clause

Empno	ename	Sal	Deptno
1	Scott	1800	30
2	Jones	1950	20
4	Adams	3000	10
6	Smith	2750	30
7	Miller	2000	20
8	Kings	3500	30

Output of having clause

1	Scott	1800	30
6	Smith	2700	30
8	King	3500	30

2	Jones	1950	20
7	Miller	2000	20

Output of group by clause

30

1	Scott	1800	30
6	Smith	2750	30
8	King	3500	30

20

2	Jones	1950	20
7	Miller	2000	20

10

4	ADAMS	3000	10

group filter condition count (*) >= 2

30

3 >= 2 (T)

20

2 >= 2 (T)

10

1 >= 2 (F)

Count (*)	Deptno
3	30
2	20

output of select clause

(Exectly, Atleast)

* Having clause : It is used to filter the groups
→ without group by clause we can not use having clause.
→ having clause execute after the group by.
→ In having clause we can use group filter condition as a argument. (we can use multi row functions as a group filter condition).
→ having clause executes group-by-group.

Q1 WAQTD No. of employees earning sal greater than 1500 Atleast 2 employee are Working in each department.

⇒
group function ← group by expression
(5) select count(*), Deptno
(1) From emp
(2) Where sal > 1500 → filter condition
(3) Group by deptno → column name
(4) Having count(*) >= 2;
group filter condition
MRF()

Q2 WAQTD maximum salary given to an employees atleast 4 employees are working in each job.

⇒ select max(sal), job
from emp
group by job
Having count(*) >= 4;

Q3. WAQTD total salary and no. of employees exctly 3 employes are working as manager or salasemen in each job.

2) select sum(sal), count(*), job
from emp
where JOB IN ('MANAGER', 'SALESMAN')
group by job
Having count(*) = 3;

Q4 WAQTD Maximum salary given to An employe in each job the maximum salary is less than 2000.

⇒ select max(sal), job
from emp
~~where sal<2000~~
group by job
Having ~~count~~ max(sal) < 2000;

Q5. WAQTD maximum salary and minimum salary given to employees in each deptment minimum salary is less than 1000 and maximum salary greate then 2000.

⇒ select max(sal), min(sal), deptno
from emp
group by deptno
having ~~max~~ min(sal) < 1000 AND max(sal) > 2000;

Q6. WAQTD hiredates which are repeted in a employe table.

⇒ select Hiredate
from emp

group by Hiredate
Having count (*) > 1;

Q7 # ORDER BY CLAUSE #

→ Order by clause is used to arrange the records either in ascending order or in decending order.
→ By default Order By clause sorts the records in ASC order.
→ Order By clause executed at the last.
→ Order by clause executes after select clause.
→ Order by clause executes ROW BY ROW.

Syntax :-

```
SELECT group_by_expression/group function
FROM table_name
[WHERE <filter condition>]
[GROUP BY column_name/expression]
[HAVING <group_filter_condition>]
 ORDER BY colname/expression [ASC]/DESC; DESC;
```

ORDER OF EXECUTION

1.	FROM	
2.	WHERE (if used)	[ROW-BY-ROW]
3	GROUP BY (if used)	[ROW-BY-ROW]
4	HAVING (if used)	[GROUP-BY-GROUP]
5	SELECT	[GROUP-BY-GROUP]
6	ORDER BY	[ROW-BY-ROW]

Page No.
Date / /

Q 1. WAQTD sal in Assending order.

⇒ Select sal
from emp
order by sal;

Q2. WAQTD name and deptno of employees if the employees are coworking in deptno 10 & 20 arrenge the employee names in decending order.

⇒ Select ename, deptno
from emp
where deptno. IN (10, 20)
Order by ename desc;

Q3 WAQTD job and salary of an employee if the employees are earning 4 digit salary arrenge the salary in a descending order.

⇒ Select job, sal
from emp
where ~~ename~~ sal LIKE '----';
Order by sal desc;

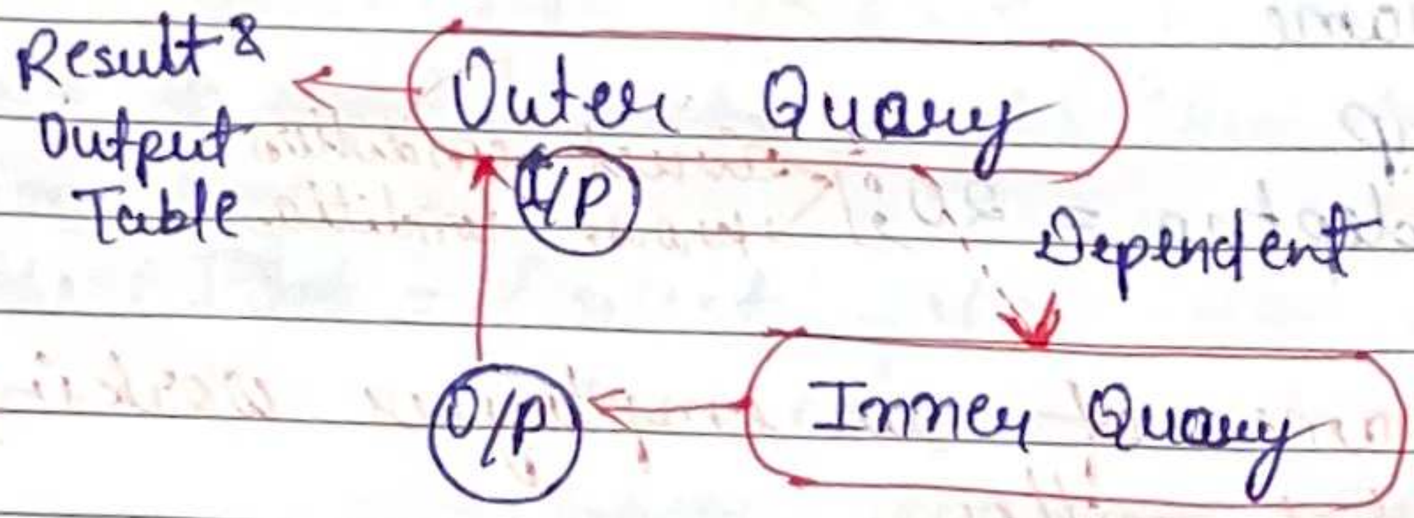
Q4 WAQTD number of employees, job if the employee are hire during in the '81 in each job. arrenge the job column in a assending order.

⇒ Select count (*), job
from emp
where hiredate Between '01-JAN-81' AND '01-DEC-81'
group by job
Order by job;

Sub-Query

Page No.
Date

Q5 A Query written inside another query is known as subquery.



→ In subquery let as consider two query outer query and inner query.
→ In subquery inner query will execute first and generate an output.
→ The output of a inner query is taken as a input to the outer query. query.
→ with the help of this input outer query will execute and generates an output.
→ Therefore Outer query is depending on inner query.

* When and why we use subqueries. queries.

1. [case:1] - When There is a unknown condition present in a queries we use subqueries

emp

Ename	Sal	Dept
ADAMS	1800	30
JAMES	3500	10
MILLER	1950	30
BLAKE	1500	20
KING	4000	10

Page No
Date

Q 1 WAQTD name of employees who are working in deptno 20.

⇒ Select ename
from emp
where [deptno = 20;] ← Direct condition / known condition

Q 2 WAQTD names of an employees working in the Same dept of miller.

⇒ Select ename
from emp
where Deptno = 'MILLER';
Indirect condition
or
Unknown condition

[ADAMS
MILLER]

Select ename
from emp
where Deptno = (select Deptno
from emp
where ename = 'MILLER');

30 (O/P)

30	(T)
10	(F)
30	(T)
20	(F)
10	(F)

MILLER	1950	30

Q.3 WAQTD details of the employees who are earning Salary Less the Blacke.

⇒ select *
from emp
where sal < (select sal
from emp
where ename = 'BLAKE');

Page No.
Date

Q4. WAQTD Details of employees along with annual salary if the employees are working in the same designation of 'JAMES'.

⇒
```
select emp.*, sal*12 AS "Annual Salary"
from emp
where Job = (select job
             from emp
             where ename = 'JAMES');
```

Q5 WAQTD name and hiredate of employees if the employees are hire after miller.

⇒
```
select ename, hiredate
from emp
where hiredate > (select hiredate
                  from emp
                  where ename = 'MILLER');
```

Q6 WAQTD employee name and annual salary of an employees if the employee are earning annual salary less than the 'scott'.

⇒
```
select ename, sal*12 AS "Annual Sal"
from emp
where sal*12 < (select sal*12
                from emp
                where ename = 'SCOTT');
```

Q7 WAQTD details of an employees who are working as a manager of or Analyst is the same dept. of adams.
Direct Indirect

⇒
```
select *
from emp
where job IN ('MANAGER', 'ANALYST') AND Deptno =
```

(select Deptno
from emp
where ename = 'ADAMS');

Q 8 WAQTD name and commision of an employes if the employes are hire after 'ALLEN' and earning some commission.

⇒ select ename, comm
from emp → comm is NOT NULL AND
where hiredate > (select hiredate
from emp
where ename = 'ALLEN') ~~AND~~
~~comm IS NOT NULL~~);

Q 9. WAQTD ename of employes if the employes are working in the same dept of jones and hire before Scott.

⇒ select ename,
from emp
where deptno = (select deptno
from emp
where ename = 'JONES') AND
hiredate < (select hiredate
from emp
where ename = 'SCOTT');

Q 10 WAQTD name and salary of an employes who are earning salary greater than smith and less than jones.

⇒ select ename, sal
from emp
where sal > (select sal
from emp

where ~~sal~~ ename = 'SMITH') AND
sal < (select sal
from emp
where ename = 'JONES');

Q11 WAQTD employee name, sal, hiredate of an employee if the employees are working same Designation of Turner and earning Salary less than the KING.

⇒ Select ename, sal, hiredate
from emp
where job = (select job
from emp
where ename = 'TURNER') AND
sal < (select sal
from emp
where ename = 'KING');

Assignment — SUB Query CASE I — 08

Q.1 WAQTD Name of the employees earning more than ADAMS

⇒ select ename
from emp
where sal > (select sal
from emp
where ename = 'ADAMS');

Q2 WAQTD Name and salary of the employee earning Less than King.

⇒ select ename, sal
from emp

where ~~sal~~ ename = 'SMITH') AND
sal < (select sal
from emp
where ename = 'JONES');

Q11 WAQTD employee name, sal, hiredate of an employee if the employees are working same Designation of Turner and earning Salary less than the KING.

⇒ select ename, sal, hiredate
from emp
where job = (select job
from emp
where ename = 'TURNER') AND
sal < (select sal
from emp
where ename = 'KING');

Assignment — SUB Query CASE 1 — 08

Q.1 WAQTD Name of the employees earning more than ADAMS

⇒ select ename
from emp
where sal > (select sal
from emp
where ename = 'ADAMS');

Q2 WAQTD Name and salary of the employee earning Less than King.

⇒ select ename, sal
from emp

where sal <= (select sal
from emp
where Ename = 'KING');

Q3 WAQTD name and deptno of employee if they are working in the same dept as JONES.

⇒ select ename, deptno
from emp
where deptno = (select deptno
from emp
where ename = 'JONES');

Q4 WAQTD Name and JOB of all the employee working in same Designation as James.

⇒ select ename, job
from emp
where job = (select job
from emp
where ename = 'JAMES');

Q5 WAQTD EMPNO AND ENAME along with annual salary of all employee if their annual salary is greater than WARDS Annual salary

⇒ select empno, ename, sal * 12 AS "Annual Sal"
from emp
where sal*12 > (select sal*12
from emp
where ename = 'WARD');

Q6. WAQTD NAME AND HIREDATE of the employee if they are hired Before SCOTT.

⇒ select ename, hiredate
from emp
where hiredate < (select hiredate
from emp
where ~~hiredate~~ ename = 'SCOTT');

Q7 WAQTD name and hiredate of the employees if they are hired after the president.

⇒ Select ename, hiredate
from emp
where hiredate > (select hiredate
from emp
where job = 'PRESIDENT');

Q8. WAQTD Name and sal of the employee if they are earning sal less than the employee whos empno is 7839.

⇒ select ename, sal
from emp
where sal < (select sal
from emp
where empno = 7839);

Q9. WAQTD all the details of the employees if the employees are hired Before MILLER.

⇒ Select *
from emp
where hiredate < (select hiredate
from emp
where ename = 'MILLER');

Page No.
Date / /

Q10. WAQTD ename and empno of the employees if employees are earning more than ALLEN.

→ select ename, empno
from emp
where sal > (select sal
from emp
where ename = 'ALLEN');

Q11. WAQTD Ename and salary of all the employees who are earning more than Miller. But Less than ALLEN.

→ select ename, sal
from emp
where sal > (select sal
from emp
where ename = 'MILLER') AND
sal < (select sal
from emp
where ename = 'ALLEN');

Q12 WAQTD all the details of the employees working in dept 20 and working in the same Designation as SMITH.

→ select *
from emp
where ~~job = (select job~~
~~from emp~~
~~where~~ deptno = 20 AND job =
(select job
from emp
where ~~job~~ ename = 'SMITH');

Page No.
Date

Q13 WAQTD all the details of the employees working as manager is the same dept as TURNER.

⇒
```
select *
from emp
where job = 'MANAGER' AND deptno = (select deptno
                                    from emp
                                    where ename = 'TURNER');
```

Q14 WAQTD Name and hiredate of the employees hired after 1980 And Before KING.

⇒
```
select ename, hiredate
from emp
where hiredate > '31-DEC-1980' AND hiredate <
                (select hiredate
                from emp
                where ename = 'KING');
```

Q15 WAQTD name and sal along with annual sal for all Employees who's sal is less than Blake and more than 3500.

⇒
```
select ename, sal, sal * 12 AS "Annual_Sal"
from emp → sal > 3500 AND
where sal < (select sal
             from emp
             where ename = 'BLAKE');
```

Q16 WAQTD All the details of employee who earn more than Scott but Less than King.

⇒
```
select *
from emp
where sal > (select sal
             from emp
```

where ename = 'SCOTT') AND sal <
(select sal
from emp
where ename = 'KING');

Q17 WAQTD name of the employees who's name starts with 'A' and working in the same dept as BLAKE.

⇒ select ename
from emp
where ename LIKE 'A%' AND deptno =
(select deptno
from emp
where ename = 'BLAKE');

Q18 WAQTD name and comm if employee earn commision and work is the same designation as SMITH.

⇒ select ename, comm
from emp
where comm IS NOT Null AND job = (select job
from emp
where ~~job~~ ename =
'SMITH');

Q19 WAQTD details of all the employees working as clerk in the same dept as TURNER.

⇒ select *
from emp
where job = 'CLERK' AND Deptno = (select Deptno
from emp
where ename = 'TURNER');

Page No.
Date

Q20 WAQTD ename, sal and Dessignation of the employee who's annual salary is more than smit and less than king.

→ select ename, sal, job
from emp
where sal*12 > (select sal*12
from emp
where ename = 'SMITH') AND
sal*12 < (select sal
from emp
where ename = 'KING');

* CASE 2: Data to be selected and condition to executed are present in ~~deferen~~ different table we use sub query.

Emp

ename	sal	Deptno
JAMES	1800	30
BLAKE	1750	10
FORD	2500	20
ADAMS	1250	10
KING	3700	20

Dept

Deptno	Dname	Loc
10	D1	L1
20	D2	L2
30	D3	L3

Q. WAQTD name of the employees who are earning sal is 2500

* select ename → Data to be store
from emp
where sal = 2500; → condition

Same ~~or~~ table

Q. WAQTD deptname of an employees if the employees names is ADAMS.

* select Dname → Data to be selected
from Dept
where ename = 'ADAMS';
↓ condition
condition belongs to different table

O/P → [D1]

* select Dname → Data to be selected
from Dept
where Deptno = (select Deptno
from emp
where ename = 'ADAMS');

→	10	(T)
→	20	(F)
→	30	(F)

10 (O/P)

condition

ADAMS	1250	10

Q1 WAQTD dept name and location of the employees if the employee name smith.

→ select dname, loc
from dept
where Deptno = (select Deptno
from emp
where ename = 'SMITH');

Q2 WAQTD dept details of of the employees if the employees are hire during the year 82.

→ select *
from dept
where Deptno = (select Deptno
from emp
where hiredate LIKE '%82');

Page No.
Date

Q3 WAQTD Deptname and Dept no. of the employee if the employees are earning salary as 800.

⇒
```
select Dname, Deptno
from Dept
where Deptno = (select Deptno
                from emp
                where sal = 800);
```

Q4. WAQTD names of an employees who are working in the dept of accounting

⇒
```
select ename
from emp
where deptno = (select Deptno
                from Dept
                where Dname = 'ACCOUNTING');
```

Q5. WAQTD no. of employees working in the location of New York.

⇒
```
select count (*)
from emp
where Deptno = (select Deptno
                from Dept
                where loc = 'NEW YORK');
```

Q6. WAQTD details of an employees working as a analyst in the dept of research

⇒
```
select *
from emp
where job = 'ANALYST' AND Deptno = (select Deptno
                                    from Dept
                                    where Dname =
                                    'RESEARCH');
```

Page No.
Date

Q.7 WAQTD name, and hiredate of employees if the employees are hire before the year 85 and working until the location CHICAGO

→ select ename, hiredate
from emp
where hiredate < '01-JAN-85' AND Deptno =
(select Deptno
from Dept
where LOC = 'CHICAGO');

Q8. WAQTD details of an employees if the employees are hire after allen and working the the location of NEW YORK

→ select *
from emp
where hiredate > (select hiredate
from emp
where ename = 'ALLEN')
AND Deptno = (select Deptno
from ~~location~~ Dept
where loc = 'NEW YORK');

Q9: WAQTD Details of employees if the employees are working in the same Designation of TURNER and working in the location CHICAGO.

→ select *
from emp
where job = (select job
from emp
where ename = 'TURNER') AND
Deptno = (select Deptno
from Dept

where LOC = 'CHICAGO');

Q10 HAQTD Details of employees along with annual salary if the employees are earning annual salary greater than MARTIN and less than king and working in the Dept of RESEARCH

⇒ Select emp.*, sal*12 AS "Annual-sal"
from emp
where sal*12 > (select sal*12
from emp
where ename = 'MARTIN') AND
sal*12 < (select sal*12
from emp
where ename = 'KING') AND
Deptno = (select Deptno
from Dept
where Dname = 'RESEARCH');

Assignement - 09

Q21 HAQTD Dname of the employees who's name is smith.

⇒ select Dname
from ~~emp~~ Dept
where Deptno = (Select Deptno
from ~~Dept~~ emp
where ename = 'SMITH');

Page No
Date / /

Q2 WAQTD Dname and Loc of the employee who's ename is king.

⇒
```
select dname, loc
from Dept
where Deptno = (select Deptno
from emp
where ename = 'KING');
```

Q3 WAQTD LOC of the emp who's employee number is 7902

⇒
```
select LOC
from Dept
where Deptno = (select Deptno
from emp
where empno = 7902);
```

Q4 WAQTD Dname and loc along with Deptno of the employee whoes name ends with R.

⇒
```
select Dname, loc, Deptno
from Dept
where Deptno IN (select Deptno
from emp
where ename LIKE '%R');
```

Q5 WAQTD Dname of the employee whos Designation is president.

⇒
```
select Dname
from Dept
where Deptno = (select Deptno
from emp
where job = 'PRESIDENT');
```

Q6 WAQTD Name of the employee working in accounting Departement.

⇒
```
select Ename
from emp
where Deptno = (select Deptno
from Dept
where Dname = 'ACCOUNTING');
```

Q7 WAQTD Ename and salaries of the employees who are working in the location 'Chicago.

⇒
```
select Ename , Sal
from emp
where Deptno = (select Deptno
from Dept
where LOC = 'CHICAGO');
```

Q8 WAQTD Details of the employees working in sales

⇒
```
select *
from emp
where Deptno = (select Deptno
from Dept
where Dname = 'SALES');
```

Q9 WAQTD Details of the employee along with annual salary if employees are working in NEW YORK.

⇒
```
select *
from emp
where Deptno = (select Deptno
from Dept
where LOC = 'NEW YORK');
```

Page No
Date / /

Q 10 WAQTD name of employees working in operations Departement.

→ select ename
from emp
where Deptno = (select Deptno
from Dept
where Dname = 'OPERATIONS');

Q 11 WAQTD name of the employee earning more than SCOTT in Accounting Dept.

→ select ename
from emp
where sal > (select sal
from emp
where ename = 'SCOTT') AND Deptno = (select Deptno
from Dept
where Dname = 'ACCOUNTING');

Q 12 WAQTD Details of the employees working as manager in the location Chicago.

→ select *
from emp
where job = 'MANAGER' AND Deptno = (select Deptno
from Dept
where loc = 'CHICAGO');

Q 13 WAQTD name and sal of the employee earning more than king in the Dept accounting.

→ select ename, sal
from emp
where sal > (select sal

from emp
where ename = 'KING') AND Deptno = (select Deptno
from Dept
where Dname = 'ACCOUNTING');

Q14 WAQTD Details of the employee working as salesman in the Department SALES.

⇒ select *
from emp
where job = 'SALESMAN' AND Deptno = (select Deptno
from Dept
where Dname = 'SALES');

Q15 WAQTD Name, sal, job, Hiredate of the employee working in operation Department and hiredate Before King.

⇒ select ename, sal, job, hiredate
from emp
where hiredate < (select hiredate
from emp
where ename = 'KING') AND Deptno = (select Deptno
~~whe~~ from Dept
where Dname = 'OPERATION');

Q16 Display all the employees whoes Department Names ENDing 'S'.

⇒ select ename
from emp
where Deptno = (select Deptno
from Dept
where Dname LIKE '%S');

Page No.
Date / /

Q17 WAQTD Dname of the employee whose name has character A In it.

→ select Dname
from Dept
where Deptno = (select Deptno
from emp
where ename LIKE '%A%');

Q18 WAQTD Dname and loc of the employee who's salary is rupees 800

→ select Dname
from Dept
where Deptno = (select Deptno
from emp
where sal = 800);

Q19 WAQTD Dname of the employee who earn commision

→ select Dname
from Dept
where Deptno = (select Deptno
from emp
where comm IS NOT NULL);

Q20 WAQTD LOC of the employee if they earn commision in dept 40

→ select LOC
from Dept
where Deptno = (select Deptno
from emp
where comm IS NOT NULL AND Deptno = 40);

05/05/25

Page No.
Date

Types of Sub-Queries:

1. Single row Sub Query
2. Multi row Sub Query

1. Single row Sub query:

→ If the sub query or inner which return exactly only one output is known as Single row sub query.
→ For single row subqueries we can use normal operators (=, !=, <, >, <=, >=) or special operator (IN, NOT IN)

Q. WAQTD names of an employees who are working in the same department of ALLEN.

⇒ Emp

ename	sal	Deptno
ADAMS	1800	20
JONES	3000	10
BLAKE	1850	20
ADAMS	1250	30
ALLEN	950	10

[JONES, ALLEN]

```
Select ename
from emp
where Deptno = (select Deptno
                from emp
                where ename = 'ALLEN');
```

10 (O/P)

→ 20 (F)
→ 10 (T) ← [ALLEN 950 10]
→ 20 (F)
→ 30 (F)

2. Multi Row Sub Query:

→ If sub query or inner query which return more than one output or multiple of output is known as multi Row Sub query.

→ for multi row sub query we can't use normal operators.

→ So we can use special operator for sub query operator.

Special operator — IN, NOT IN

Subquery operator = ANY, ALL

Q Write a query to display name of an employees who are working in the same Dept of ~~add~~ ADAMS.

ADAMS
BLAKE
ADAMS

select ename
from emp
where Deptno (=) (select Deptno (20,30) (O/P)
from emp
where ename = 'ADAMS');

→	20	(T) ✓
→	10	(F)
→	20	(T) ✓
→	30	(T) ✓
→	10	(F)

IN

ADAMS	1800	20
ADAMS	1250	30

Q NOTE

It is deficault of identify the query belongs to single row Subquery or multi row Subquery so we cannot use any normal operator so we can use only special operators in Subquery

For ex: For the case of equals we go with IN operator. For the case of not equals we go with NOT IN operator

* Sub Query Operators :
1. Any
2. All

1. All operator : all operator is a subquery operator.
→ we can use all operator along with the relational operators
→ All operator returns true if All the conditions get satisfied. (Output of inner query).

→

Q : WAQTD NAMES of an employees who are earning salary greater than all the ADAMS.

⇒ [JONES BLAKE]

```
select ename
from emp
where sal (>) All (select sal
                   from emp
                   where ename = 'ADAMS');
```

>All

(1800, 1250) (dp)

ADAMS	1800	20
ADAMS	1250	30

1800 > All (1800, 1250) → F
F F

3000 > All (1800, 1250) → T ✓
T T

950 > All (1800, 1250) → F
F F

1950 > All (1800, 1250) → T ✓
T T

1250 > All (1800, 1250) → F
F F

2. **ANY Operators :-** Any operator is a subquery operator
→ we can use any operator along with relational operator.
→ Any operator returns true atleast any one of the condition get satisfied. (Atleast or any one of output of inner query get satisfied).

Q. WAQTD Names of employees who are earning sal greater than atleast any one of the ADAMS.

A)
```
select ename                [ADAMS
from emp                     JONES
where sal > Any (select sal  BLAKE]
                 from emp                (1800, 1250)
                 where ename = 'ADAMS');
```

ADAMS	1800	20
ADAMS	1250	30

1800 > Any (1800, 1250) → T
F T

3000 > Any (1800, 1250) → T
T T

1950 > Any (1800, 1250) → T
T T

1250 > Any (1800, 1250) → F
F F

950 > Any (1800, 1250) → F
F F

Assignment -10

ALL & Any

Q1 WAQTD Name of the employee earning salary more than all the salesman.

⇒
```
select ename
from emp
where sal > ALL ( select sal
from emp
where job = 'SALESMAN');
```

Q2 WAQTD Details of the employees hired after all the clerks.

⇒
```
select *
from emp
where hiredate > All (select hiredate
from emp
where job = 'CLERK');
```

Q3 WAQTD name and salary for all the employees if they are earning less than atleast a manager.

⇒
```
select ename , sal
from emp
where sal < ANY (select sal
from emp
where job = 'MANAGER');
```

Q4 WAQTD name and hiredate of employees hired before all the managers.

⇒
```
select ename, hiredate
from emp
where hiredate <ALL (select hiredate
from emp
where job = 'MANAGER');
```

Page No.
Date

Q5 WAQTD name of the employee hired after all the managers and earning salary more than all the clerks.

⇒ select ename
from emp
where hiredate > All (select hiredate
from emp
where job = 'MANAGERS') AND Sal > All(select
Sal from emp
where clerks job = 'CLERKS');

Q6. WAQTD Details of the employee working as Clerk and hired Before atlest a salesman.

⇒ select *
from emp
where job = 'CLERK' AND hiredate < any
(select hiredate
from emp
where job = 'SALESMAN');

Q7 WAQTD Details of employees working in accounting or Sales Dept.

⇒ select *
from emp
where ~~job~~ Deptno IN (select Deptno
from ~~emp~~ Dept
where ~~job~~ Dname IN ('ACCOUNTING', 'SALES');

Q8 WAQTD Department name of the employees with name Smith, King AND MILLER.

⇒ Select Dname
from Dept

where Deptno ~~IN~~ (select Deptno
from ~~emp~~ emp
where ename IN ('SMITH', 'KING', 'MILLER'));

Q9 WAQID Details of employees working NEW YORK OR CHICAGO.

→ select *
from emp
where Deptno IN (Select Deptno
from Dept
where LOC IN ('NEW YORK', 'CHICAGO'));

Q10 WAQTD Emp name if employee are hired after all the employees of dept 10.

→ Select ename
from emp
where hiredate > All (select hiredate
from emp
where Deptno = 10);

Maximum and minimum use in Sub query

Q.1: Write a query to Display name of employees getting maximum salary.

Emp

ename	sal
JONES	1850
BLAKE	3000
ADAMS	1500
MILLER	2750
FORD	1950
KING	2000

```
select ename
from emp
where sal IN (select max(sal)
from emp);
```

(3000)

sal	
1850	F
3000	T
1500	F
2750	F
1950	F
2000	F

Q 2 WAQTD details of employees who earning minum Salary

```
select *
from emp
where sal IN (select min(sal)
from emp);
```

Q 3 WAQTD name and hiredate of an employees who are hire at the last.

```
select ename, hiredate
from emp
Where hiredate IN (select max(hiredate)
from emp);
```

Q 4 WAQTD details of employees if the employees are having least of emptaye no.

```
select *
from emp
where empno IN (select min(empno)
from emp);
```

NOTE Last → max()
      First → min()

Page No.
Date

Q5. WAQTD hiredates of an employee : who are hire at the first.

=> select hiredate
from emp
where hiredate IN (select min(hiredate)
from emp);

* NESTED SUBQUERY

→ A Subquery written inside another query is known as nested subquery.
→ We can nest upto 255 sub query.

NOTE :

max()	<
min()	>

Q. WAQTD 1st maximum salary from employee table.

=> select max(sal) (3000) 1st max salary
from emp;

Q. WAQTD 2nd maximum salary form employee.

=> select max(sal) 2750 2nd max sal
from emp
where sal < (select max(sal) (3000) 1st max salary
from emp);

1850 T
3000 T
1500 T
2750 T ✓
1950 T
3000 F X

Q:- WAQ TD 3rd max salary from employee table.

(2000) 3rd sal

```
Select max(sal)
from emp
where sal < (Select max(sal)
from emp
where sal < (select max(sal)
from emp));
```

sal	
1850	T
3000	F
1500	T
2750	F
1950	T
2000	T

(2750) 2nd sal

(3000) 1st sal

sal	
1850	T
3000	F
1500	T
2750	T
1950	T
2000	T

Q. WAQTD 4th max salary from employee table.

(1950) 4th max sal

```
Select max(sal)
from emp
where sal<(select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp)));
```

sal	
1800	T
3000	F
1500	T
2750	F
1950	F
2000	F

(2000) 3rd sal

sal	
1850	T
3000	F
1500	T
2750	F
1950	T
2000	T

(2750) 2nd sal

sal	
1850	T
3000	F
1500	T
2750	T
1950	T
2000	T

(3000) 1st sal

sal	
1850	T
3000	F
1500	T
2750	T
1950	T
2000	T

(Minimum)

Q.1 WAQTD 2nd min sal from employee table

```
select min(sal)
from emp
where sal > (select min(sal)
             from emp);
```

Q.2 WAQTD names of an employees, who are earning 3rd maximum salary.

```
select ename
from emp
where sal IN (select max(sal)
              from emp
              where sal < (select max(sal)
                           from emp)));
```

Q3 WAQTD details of an employees who are hire before the last employee.

```
select *
from emp
where hiredate IN (select max(hiredate)
from emp
where hiredate < (select max(hiredate
from emp));
```

Q4 WAQTD departement name of employ who are earning second min sal.

```
select Dname
from Dept
where Deptno IN(select Deptno
```

```
from emp
where sal IN (select min(sal)
from emp
where sal > (select min(sal)
from emp)));
```

Q.5 WAQTD Department name and location of the employee who are earning 5th maximum salary

```
select Dname, loc
from Dept
where Deptno IN (select Deptno
from emp
where sal IN (select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp)))));
```

Assignment

Page No.
Date / /

Q1 WAQTD name of the employee earning maximum salary.

⇒
```
select ename
from emp
where sal IN(select max(sal)
from emp);
```

Q2 WAQTD name of the employee earning minimum sales

⇒
```
select ename
from emp
where sal IN (select min(sal)
from emp);
```

Q3 WAQTD name and hiredate of the employee fired at the last.

⇒
```
select ename, hiredate
from emp
where hiredate IN (select max (hiredate)
from emp);
```

Q4 WAQTD Name and Hiredate of the employee Hired at the first.

⇒
```
Select ename, hiredate
from emp
where hiredate IN (select min (hiredate)
from emp);
```

Q5 WAQTD name, comm of the employee who earn min commision

⇒
```
select ename, comm
from emp
where comm IN (select min (comm) from emp);
```

Q6 WAQTD name, sal and comm of the employee earning maximum commission

⇒
```
select ename, sal, comm
from emp
where comm IN (select max(comm)
from emp);
```

Q7 WAQTD Details of the employee who has greater empno.

⇒
```
select *
from emp
where empno IN (select max(empno)
from emp);
```

Q8 WAQTD Details of the employee having ~~last~~ least Hiredate.

⇒
```
select *
from emp
where hiredate IN (select min(hiredate)
from emp);
```

Q9 WAQTD Details of the employee earning least annual salary.

⇒
```
select *
from emp
where sal*12 IN (select min(sal*12)
from emp);
```

Q10 WAQTD name, annual salary of the employee if their annual salary is more than all the Salesman.

Page No.
Date

```
select ename, sal*12 As
from emp
where sal*12 > All (select sal*12
from emp
where job = 'SALESMAN');
```

Q1 NESTED Sub Query
Q1 WAQTD 2nd minimum salary.

```
select min(sal)
from emp
where sal > (select min(sal)
  from emp);
```

Q2 WAQTD 5th maximum salary.

```
select max(sal)
from emp
where sal < (select max(sal)
from emp
 where sal < (select max(sal)
 from emp
 where sal < (select max(sal)
 from emp
 where sal < (select max(sal)
 from emp ))));
```

Q3 WAQTD Name of the employee earning 3rd maximum salary.

```
select ename
from emp
where sal IN (select max(sal)
```

from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp)));

Q4 WAQTD EMPNO of the employee earning 2nd maximum Salary.

⇒ select empno
from emp
where sal IN (select max(sal)
from emp
where sal < (select max(sal)
from emp));

Q5 WAQTD Department name of an employee getting 4th max sal

⇒ select Dname
from emp
where Deptno IN (select Deptno
from emp
where sal IN (select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (select max(sal)
from emp
where sal < (Select max(sal)
from emp)))));

Page No.
Date

Q6 WAQTD Details of the employee who was hired 2nd

⇒ select *
from emp
where hiredate in (seleteect min(hiredate)
from emp
where hiredate > (select min(hiredate)
from emp));

Q7 WAQTD Name of the employee Hired before the last employee

⇒ select ename
from emp
where hiredate < (select max(hiredate)
from emp);

Q8 WAQTD loc of the employee who was hired first

⇒ select loc
from Dept
where Deptno IN (select Deptno
from emp (select
where hiredate IN min(hiredate)
from emp)));

Q9 WAQTD Details of the employee earning 7th the min salary

⇒ select *
from emp IN
where sal" (select min(sal)
from emp
where sal > (select min(sal)
from emp
where sal > (select min sal)

from emp
where sal > (select min(sal)
from emp
where sal > (select min(sal)
from emp
where sal > (select min(sal)
from emp))))));

10 WAQTD Dname of employee getting 2nd maximum salary.

⇒ select Dname
from emp Dept
where Deptno IN (select Deptno
from emp
where sal IN (select max(sal)
from emp
where sal < (select max(sal)
from emp)));

Page No.
Date

```
Select count(*), deptno
from emp
where job = 'CLERK'
group by deptno;
```

① NAME ID name of the employee if the emp. earning same sal. in Dept of 10, 20

⇒
```
Select ename
from emp
where sal IN (select sal
from emp
where Deptno IN (10,20));
```

Page No.
Date

Ans Difference B/w primary key & unique:

primary key	unique
1. Primary key can't accept null values.	1. Unique can accept null values.
2. We can have only one primary key in table.	2. We can have more than one unique key is a table
3. used to identify unique records from table.	3. ensure value in column are unique.
(4) we can not delete primary key value.	(4) we can delete unique value

Q. Difference between unique & distinct.

unique	distinct
(1) Unique can not accept duplicate and repeted value while enter the data in table	(1) Distinct reject duplicate and repeted value while facthing data.
(2) Unique accept null value.	(2) Distinct can not accept duplicate value.
(3) Unique is a constraint	(3) Distinct is keyword.

☆ Difference b/w primary key & Foreign key

☆ Difference b/w primary key & unique.

Ans Difference b/w primary key & foreign key.

primary key	foreign key
1. It is used to identify unique record from the table.	1. It is used to Establish connection between the table.
2. We have only one primary key in table.	2. We have 'n' number of foreign key in a table.
3. Primary key can't accept duplicate or repeted value	3. foreign key can accept duplicate and repeted values.
4. Primary key can't accept null values.	4. foreign key can accept null values.
5. Primary key is a combination of unique & not null constraint.	5. foreign key is a not a combination of unique and not null constraint.

Page No.
Date 09/04/24

* Employer Manager Relationship

1. CASE : 1 — To Find A Manager

EMPO IN MGR

emp

empno	Ename	MGR
1	ALLEN	3
2	BLAKE	1
3	JONES	4
4	MILLER	5
5	KING	2

Q WAQTD ALLENS Manager name.

Select ename [JONES]
from emp
where empno IN (select MGR
from emp
where ename = 'ALLEN');

3 (O/P)

→ 1 (F)
→ 2 (F)
→ 3 (T)
→ 4 (F)
→ 5 (F)

1 ALLEN 3

Q1 WAQTD Manager name of smith

⇒ select ename
from emp
where empno IN (select MGR
from emp
where ename = 'SMITH');

Q2 WAQTD salary of allen's manager

⇒
```
select sal
from emp
where empno In (select MGR
 from emp
 where ename = 'ALLEN');
```

Q3 WAQTD Jones managers manager name.

⇒
```
select ename
from emp
where empno IN (select MGR
 from emp
 where Empno IN (select MGR
 from emp
 where ename = 'JONES'));
```

Q4 WAQTD Dept name of Turners Managers.

⇒
```
select Dname
 from Dept
 where Deptno IN (select Deptno
 from emp
 where Empno IN (select MGR
 from emp
 where ename = 'TURNER'));
```

Q5 WAQTD Details of BLACKS Manager.

⇒
```
select *
from emp
 where empno IN (select MGR
 from emp
 where ename = 'BLAKE');
```

Page No.
Date

Q6.

* CASE : 2 — To find a employee / Reporting

MGR IN EMPNO

Q WAQTD NAMES of a employee reporting to miller

→ Manager

[JONES]

```
select ename
from emp
where MGR IN (select empno
              from emp
              where ename = 'MILLER');
```

4 [O/P]

→	3	(F)
→	1	(F)
→	4	(T) ✓
→	5	(F)
→	2	(F)

4	MILLER	5

Q1 WAQTD Details of employees who are reporting to BLAKE.

⇒
```
select *
from emp
where MGR IN (select empno
              from emp
              where ename = 'BLAKE');
```

Q2 WAQTD No. of employees who are reporting to SCOTT.

⇒
```
select count (*)
from emp
where MGR IN (select empno
```

from emp
where ename = 'SCOTT');

Q3 WAQTD names of an employee who are reporting to BLAKE'S Manager.

⇒ select ename
from emp
where MGR IN (select Empno
from emp
where ~~MGR~~ Empno IN (select ~~Empno~~ MGR
from emp
where ename = 'BLAKE'));

combination of case 1 & case 2

Q4 WAQTD Details of the employees who are reporting to FORD'S Managers

⇒ Select *
from emp
where MGR IN (select Empno
from emp
where empno IN (select MGR
from emp
where ename = 'FORD'));

Assignment - MGR

Page No.
Date

Q1 WAQTD SMITS Reporting manager's name.

```
=> select ename
   from emp
   where MGR IN (select empno
   from emp
   where ename = 'SMITH');
```

Q2 WAQTD ADAMS MANAGER'S MANAGER NAME

```
=> Select Ename
   from emp
   where empno IN (select MGR
   from emp
   where empno In (select MGR
   from emp
   where ename = 'ADAMS'));
```

Q3 WAQTD DNAME OF JONES MANAGER.

```
=> Select Dname
   from emp
   where Deptno IN (select Deptno
   from emp
   where Empno IN (select MGR
   from emp
   where ename = 'JONES'));
```

Q4 WAQTD MILLER's Manager's salary.

```
=> Select sal
   from emp
   where Empno IN (select MGR
   from emp
   where Empno IN (select MGR
```

```
from emp
where ename = 'MILLER'));
```

Q5 WAQTD LOC of Smith's Manager's manager.

⇒
```
select loc
from Dept
where Deptno IN (select Deptno
from emp
where Empno IN (select MGR
from emp
where Empno IN (select MGR
from emp
where ename = 'SMITH')));
```

Q6 WAQTD NAME of the employees reporting to Blake.

⇒
```
select ename
from emp
where MGR IN (select empno
from emp
where ename = 'BLAKE');
```

Q7 WAQTD Number of employee reporting To king.

⇒
```
Select count(*)
from emp
where MGR IN (select Empno
from emp
where ename = 'KING');
```

Page No.
Date / /

Q8 WAQTD Details of the employee reporting to Jones

⇒
```
select *
from emp
where MGR IN (select empno
from emp
where ename = 'JONES');
```

Q9 WAQTD ENAMES of the employee reporting to Blake's Manager.

⇒
```
select ename
from emp
where MGR IN(select empno
from emp
where Empno IN (select MGR
from emp
where ename ='BLAKE'));
```

Q10 WAQTD ~~NAME~~ & Number of employee reporting to FORD'S MANAGER.

⇒
```
select count (*)
from emp
where MGR IN (select empno
from emp
where Empno IN(select MGR
from emp
where ename ='FORD'));
```

Page No.
Date 10 04 25

Pseudo Column

→ Pseudo columns are the false columns which are present in each and every table in a database.
→ We can access pseduo columns implicitly
→ we can access pseduo column only in the form of explicitly (with the help of column name)

→ Type of pseudo column
1. Rowid
2. Rownum

Ex:
```
select rowid, rownum, emp.*
from emp;
```

1. Rowid :- Rowid is a 18 digit of Hexa decimal memory address
→ Rowid has generated at the time of Inserting the records.
→ Rowid is unique.
→ Rowid is static in nature.
→ Rowid we cand to able to delete or modify rowid.
→ It is used to identify duplicate records from the table.

Page No.
Date

Ex:

① SELECT Rowid, RowNum, Emp.*
from emp
wher Rowid = 'AAAMFPAAEAAAAAg.AAB';

② select *
from emp
where Rowid = 'AAAMFPAAEAAAAAg.AAB';

output ①

ROW ID	ROW NUM	EMPNO	ENAME	JOB
AAAMFPAAEAAAAg AAB.	1	7499	Smith	Salesman

Out Put②

Empno	Ename	JOB	MGR
7499	Smith	Salesman	

2. Row num :- Rownum is a serial number assigned to each and every record in a result table.

→ Rownum has generated at the time of execution
→ Rownum is Dynamic in nature. (it may change)
→ Rownum always starts with 1 and get incremented with 1.

Q WAQTD ^ Rownum first record from the employee table.

⇒

```
Select *
from emp
where rownum = 1;
```

EMPNO	ENAME	JOB	MGR	HIREDate
7369	Smith	Clerk	7902	17-DEC-80

Q.1 WAQTD first three records from the employee table

⇒

```
select *
from emp
where rownum > 0 AND rownum <= 3;
      Or Rownum IN (1, 2, 3)
```

Q2 WAQTD first half of records from the table

⇒

```
select *
from emp
where Rownum <= (select count (*)/2
from emp);
```

Q3 WAQTD first five record from the employee table

⇒

```
select *
from emp
where Rownum <= 5;
      Or
      Rownum Between 1 AND 5;
```

Page No.
Date 11/04/25

Q 4 WAQTD the 3rd record from the employee table.

⇒ Select *
from emp
where Rownum <= 3;

Select *
from emp
where Rownum <= 3;

1 <= 3 (T)
2 <= 3 (T)
3 <= 3 (T)
4 <= 3 (F)

Select *
from emp
where Rownum = 3;

1 = 3 (F)
⇒ No Increment

* To Rownum into static :

Emp

ename	sal
MILLER	1250
ADAMS	4000
BLAKE	1900
TURNER	1850
KING	3000

① STEP : 01

→ Assign rownume to a table & rename it as Serial_Number (SLNO) with the help of Alias.

For Example:

```
SELECT ROWNUM AS SLNO, EMP.*
FROM EMP;
```

OUTPUT

SLNO	Ename	SAL
1	MILLER	1250
2	ADAMS	4000
3	BLAKE	1900
4	TURNER	1850
5	KING	3000

(2) STEP:02 -

→ We use the result of the step 01 to find out the Decide record or perticulare records from the Table.

→ To use the result table, we must write the query in the ~~form~~ from clause of the outer query.

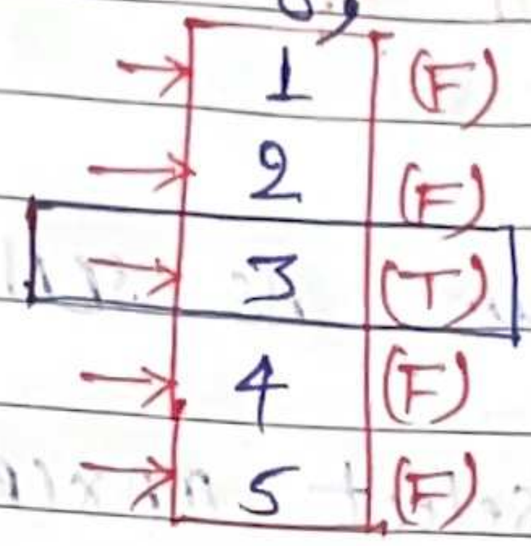
→ Then use the SLNO column as condition In the where clause.

For Example:

[3 BLAKE 1900]

```
select *
FROM (select rownum AS SLNO, Emp.*
      from emp)
```

where SLNO = 3;



Q1 WAQTD 10th record from the employee table-

⇒ Select *
from (sllect Rownum AS SLNO, emp.*
from emp)
where SLNO = 10;

Q2 WAQTD 3rd & 5th record from the employee Table.

⇒ Select *
from (select rownum AS SLNO, emp.*
from emp
wher SLNO IN(3, 5);

Q3 WAQTD last record from the employee the table.

⇒ select ~~count~~ *
from (select count(*)
from emp);

Select *
from (select Rownum as SLNO, Emp.*
from emp)
where SlNO IN (select Count(*)
from emp);

Q4 WAQTD last record of the table with using count function.

⇒
```
Select *
from (select Rownum AS SLNO , emp.*
from emp)
where SLNO IN (select MAX(Rownum)
From emp);
```

Q5 WAQTD last 3 records from the employee table.

⇒
```
select *
from (select Rownum AS SLNO, emp.*
from emp)
where SLNO >(select count(*)-3
from emp);
```

(14-3) = 11

11 > 11 (F)
12 > 11 (T)
13 > 11 (T)
14 > 11 (T)

Q6 WAQTD last five records from the employee table.

⇒
```
Select *
from (select Rownum AS SLNO, emp.*
from emp)
where SLNO > (select count(*) - 5
from emp);
```

Q7 WAQTD second half of the record from the employee table

⇒
```
select *
from (select Rownum AS SLNO, emp.*
```

from emp)
where SLNO > (select count(*)/2 (14/2) → 7
from emp);

Q8 WAQTD first three records or last three record from the employee table

select *
from (select rownum AS SLNO, emp.*
from emp)
where SLNO <=3 IN (1, 2, 3) OR
SLNO>(select cout(*) -3
from emp);

Q9 WAQTD Middel five record from the table

⇒ select *
from (select rownum as SLNO, emp.*
From emp) (14/2-2)(7-2)=(5)
Where Slno Between (select count(*)/2 -2
From emp) AND (select count(*)/2 +2
from emp); (14/2+2)(7+2)=(9)

Q10 WAQTD Even record from the employee table

⇒ Select *
from (select rownum as SLNO, emp.*
from emp)
where SLNO%2=0; where mod (Slno, 2) =0;

Q11 WAQTD ODD record from the employee table.

⇒

Page No.
Date

```
select *
from (select rownum as slno , emp.*
 from emp)
 where slno mod (slno, 2) = 1;
```